ANT-BASED VEHICLE CONGESTION AVOIDANCE FRAMEWORK USING VEHICULAR NETWORKS

MOHAMMAD REZA JABBARPOUR SATTARI

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2015

UNIVERSITI MALAYA
<u>ORIGINAL LITERARY WORK DECLARATION</u>

Name of Candidate: Mohammad Reza Jabbarpour Sattari I.C/Passport No: H95659312

Registration/Matric No: WHA12008

Name of Degree: Doctor of philosophy of Computer Science
Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Ant-based Vehicle Congestion Avoidance Framework using Vehicular Networks

Field of Study: Data Communication and Computer Networks

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before,

Witness's Signature Date

Name:

Designation:

ABSTRACT

Over the last decade, vehicle population has dramatically increased all over the world. This large number of vehicles coupled with the limited capacity of the roads and highways lead to heavy traffic congestion. Besides, it gives rise to air pollution, driver frustration, and costs billions of dollars annually in fuel consumption. Although finding a proper solution for vehicle congestion is a necessity, it is still remaining a challenging task due to the dynamic and unpredictable nature of vehicular environments. Building new high-capacity streets can be a solution but it is very costly, time consuming and in most cases, infeasible due to space limitations. However, optimal usage of the existing roads and streets capacity can lessen the congestion problem in large cities at a lower cost. Intelligent Transportation System (ITS) is a newly emerged system that aims to provide innovative services for different modes of transportation and traffic management. Vehicle Traffic Routing System (VTRS) is one of the ITS applications that can be used for efficient utilization of existing roads' capacity. Previous researches concentrated on using static algorithms to find the shortest path in VTRSs. However, providing a shortest path without considering other factors such as congestion, accidents, obstacles, travel time and speed is not a proper solution for vehicle traffic congestion problem. The efficiency of VTRSs on mitigating the vehicle congestion is challenged by the high dynamicity and quick changes of vehicular environments due to both predictable (recurring) and unpredictable (non-recurring) events. Most of the existing approaches deal with the congestion problem in a reactive manner and recover vehicle congestion implicitly, which is not a sufficient solution due to non-recurring congestion conditions. Moreover, a same path is suggested to drivers by the existing approaches which switches the congestion from one route to another, specifically, in the case of having a significant number of drivers utilizing these systems simultaneously. This research presents a bio-inspired framework, called "Ant-based Vehicle Congestion

Avoidance Framework (AVCAF)", which is a promising way to alleviate vehicle traffic congestion problem while considering the aforementioned drawbacks. AVCAF predicts vehicles' average travel speed and combines it with travel time, density, distance, map segmentation and layering to reduce congestion as much as possible by finding the least congested shortest paths in order to avoid congestion instead of recovering from it. AVCAF uses alternative paths from the early stages of the routing process. AVCAF collects real-time traffic data through vehicular networks to consider non-recurring congestion conditions in its routing mechanism via ant-based algorithm. The proposed framework is evaluated and validated through simulation environment. Experimental results conducted on three different scenarios (i.e. various vehicle densities, various system usage rates and accident condition) considering average travel time, speed, distance, number of re-routings and number of congested roads as evaluation metrics. The results show that AVCAF outperforms the existing approaches in terms of average travel time, travel speed, number of re-routings and number of congested roads.

ABSTRAK

Lebih sedekad yang lalu, populasi kenderaan telah meningkat secara mendadak di seluruh dunia. Penambahan bilangan kenderaan ditambah pula dengan keupayaan yang terhad di jalan raya dan lebuh raya menyebabkan kesesakan lalu lintas yang tinggi. Selain itu, ia menimbulkan pencemaran udara, kekecewaan pemandu, dan kos yang berbilion setiap tahun dalam penggunaan bahan api. Walaupun mencari penyelesaian yang sesuai untuk kesesakan kenderaan adalah satu keperluan, ia masih kekal sebagai tugas yang mencabar kerana sifat dinamik dan keadaan tidak dapat diramalkan daripada persekitaran kenderaan. Membangunkan jalan-jalan berkapasiti tinggi baru, penyelesaian ini adalah sangat mahal, memakan masa dan dalam kebanyakan kes, tidak dapat dilaksanakan kerana ruang yang terbatas. Sebaliknya, penggunaan optimum jalan raya dan jalan-jalan kapasiti yang sedia ada boleh mengurangkan masalah kesesakan di bandar-bandar besar pada kos yang lebih rendah. Sistem Pengangkutan Pintar (ITS) adalah satu sistem yang baru muncul bertujuan menyediakan perkhidmatan yang inovatif yang berkaitan kepada pelbagai jenis pengurusan pengangkutan dan lalu lintas. Sistem Routing Lalulintas Kenderaan (VTRS) adalah salah satu aplikasi ITS yang boleh digunakan bagi penggunaan kapasiti jalan sedia ada yang lebih efisyen. Kajian lepas ditumpukan untuk menggunakan algoritma statik untuk mencari jalan yang paling pendek dalam VTRSs. Walau bagaimanapun, menyediakan jalan yang singkat tanpa mengambil kira faktor-faktor lain seperti kesesakan, kemalangan, halangan, masa perjalanan dan kelajuan adalah bukan penyelesaian bagi kesesakan lalu lintas kenderaan yang baik. Keberkesanan VTRSs di dalam mengurangkan kesesakan lalu lintas adalah satu cabaran yang melibatkan dinamik yang tinggi dan perubahan persekitaran kenderaan disebabkan oleh peristiwa penentuan (berulang) dan tidak penentuan (tidak berulang). Kebanyakan pendekatan yang sedia ada menangani masalah kesesakan dengan cara yang reaktif dan mendapatkan kesesakan kenderaan tersirat, yang masih tidak mencukupi dalam penyelesaian kerana keadaan kesesakan tidak berulang. Selain

itu, jalan yang sama dicadangkan kepada pemandu dengan pendekatan yang sedia ada dan sama dengan algoritma statik, mereka hanya akan beralih kesesakan laluan dari satu kepada yang lain, khususnya, dalam kes yang mempunyai sejumlah besar pemandu menggunakan sistem ini pada masa yang sama. Dalam kajian ini, kami membentangkan satu rangka kerja bio-inspirasi, dipanggil rangka kerja Kenderaan Kesesakan Pengelakan (AVCAF) berasaskan "semut", yang merupakan cara yang menjanjikan untuk menyelesaikan kelemahan yang dinyatakan di atas dan mengurangkan lalu lintas kenderaan masalah kesesakan. AVCAF meramalkan kelajuan perjalanan purata kenderaan dan menggabungkan ia dengan masa perjalanan, ketumpatan, jarak, peta segmentasi dan lapisan untuk mengurangkan kesesakan sebanyak mungkin dengan mencari-kurangnya sesak laluan terpendek untuk mengelakkan kesesakan dan bukannya pulih daripadanya. AVCAF menggunakan laluan alternatif dari peringkat awal proses routing. AVCAF mengumpul data trafik masa nyata melalui rangkaian kenderaan untuk mempertimbangkan kesesakan tidak berulang dalam mekanisme penghalaan melalui algoritma semut berasaskan. Rangka kerja yang dicadangkan itu dinilai dan disahkan melalui simulasi persekitaran. Hasil ujikaji yang dijalankan ke atas tiga senario yang berbeza (iaitu ketumpatan berlainan kenderaan, pelbagai kadar penggunaan sistem dan keadaan kemalangan) mempertimbangkan masa perjalanan purata, kelajuan, jarak, bilangan penghalaan semula dan beberapa kesesakan jalan raya seperti metrik penilaian. Keputusan yang diperolehi menunjukkan bahawa AVCAF lebih baik pendekatannya dari yang sedia ada dari segi masa perjalanan purata, kelajuan perjalanan, beberapa penghalan semula dan beberapa kesesakan jalan raya.

ACKNOWLEDEMENTS

First and foremost I would like to thank God for giving me the power to believe in my passion and pursue my dreams. I could never have done this without the faith I have in you, the Almighty.

I am heartily thankful to my supervisor, Dr. Rafidah Md Noor, whose unwavering guidance, encouragement and supervision from the preliminary to the concluding level enabled me to develop an understanding of this research area. It is my privilege to have a highly dedicated supervisor like her. She has helped refine my skills in solving research problems, and in communicating research ideas to a wide range of audience.

I also thank the Ministry of Higher Education Malaysia for providing scholarship support, High Impact Research (HIR) Grant (UM.C/625/1/HIR/MOHE/FCSIT/09). This thesis would not have been possible without this support. I would also like to thank University of Malaya and all people that have helped me throughout my study.

It is my honor to have Professor Leon Rothkrantz's supports in a number of ways, especially in sharing his ideas and providing insightful comments in regards to my research.

I am indebted to my parents for their support. Without them I would not have been able to persevere during my research and complete this dissertation.

Last but not least, I would like to express my deepest gratitude to my soul mate, my friend, my wife Masumeh for her continues support and motivation ever since we met until today. I treasure every moment of my life with her and I dedicate this thesis to her.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimization
ACPre	Ant Colony Prediction
ACPs	Ant Colony Parameters
ACS	Ant Colony Segmentation
API	Application Program Interface
AS	Ant System
AVCAF	Ant-based Vehicle Congestion Avoidance framework
AWK	alfred Aho, peter Weinberger and brian Kernighan
BANT	Backward ANT
BeeJamA	Bee-inspired Jam Avoidance
BNT	Border Nodes Table
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Current Travel Speed
DSRC	Dedicated Short Range Communication
DTA	Dynamic Traffic Assignment
FANT	Forward ANT
FCC	Federal Communication Commission
GA	Genetic Algorithm
GPS	Global Positioning System
GTNetS	Georgia Technology Network Simulator
GUI	Graphical User Interface
HTS	Historical Travel Speed
ITS	Intelligent Transportation System
MAC	Media Access Control
MANET	Mobile Ad hoc Network

MAS	Multi Agent System
MOVE	MObility model generator for VEhicular networks
NAM	Network AniMator
NCTUns	National Chio Tong University network simulator
NS-2	Network Simulator-2
OBU	On-Board Unit
OD	Origin-Destination
OSI	Open System Interconnection
OSM	OpenStreetMap
OTcl	Object Tool command language
PACO	Pure Ant Colony Optimization
PANT	Packet as ANT
PSO	Particle Swarm Optimization
PTS	Predicted Travel Speed
RFID	Radio Frequency Identification
RSU	Road Side Unit
SI	Swarm Intelligence
SNS	Staged Network Simulator
STRAW	STreet RAndom Waypoint
SUMO	Simulation of Urban MObility
Tcl	Tool command language
TCP	Transmission Control Protocol
TI	Time Interval
TIGER	Topologically Integrated Geographic Encoding and Referencing
TK	ToolKit
ToS	Table of Segment

TraCI	Traffic Control Interface
TraNS	Traffic and Network Simulator
UDP	User Datagram Protocol
V2I	Vehicle-to-Infrastructure
V2R	Vehicle-to-Roadside
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad hoc Network
VANT	Vehicle as ANT
VTRS	Vehicle Traffic Routing System
WAVE	Wireless Access in Vehicular Environment

List of Appendices

CHAPTER 1: INTRODUCTION

Over the last decade, vehicle population has dramatically increased all over the world. This large number of vehicles coupled with the limited capacity of roads and highways lead to heavy traffic congestion. Besides, it gives rise to air pollution, driver frustration, and costs billions of dollars annually in fuel consumption (Narzt, Wilflingseder, Pomberger, Kolb, & Hortner, 2010). In 2010, American people faced a lot of difficulties because of vehicle congestion which force their government to spend 101 billion dollars for purchasing extra fuel (Schrank, Eisele, & Lomax, 2012). Based on a report by Texas A&M Transportation Institute (Schrank et al., 2012), it is estimated that fuel consumption will rise to 2.5 billion gallons (from 1.9 billion gallons in 2010) with cost of 133 billion dollars in 2015. Accordingly, finding effective solutions with reasonable cost for congestion mitigation is one of the major concerns of researchers and governments in recent years. Building new high capacity roadways, expanding public transport systems, using tolled ways and ramp metering have been proposed to mitigate aforementioned problems (Putha, Quadrifoglio, & Zechman, 2012). Nevertheless, these solutions are very costly, time consuming and in most of the cases, they are hardly possible because of time and space limitations. On the contrary, optimal usage of the existent and alternative roadways capacity can lessen the congestion problem in large cities at the lower cost.

The recent advances in sensing and computing technologies enables pervasive enhancement to Intelligent Transportation System (ITS) (Dimitrakopoulos & Demestichas, 2010; Liu, Fan, Branch, & Leung, 2014). ITS is a newly emerged system that collects real-time data for congestion monitoring and enhancing the drivers comfort by using road side units (RSUs) (e.g. video cameras, radio-frequency identification (RFID) readers and induction loops) and the vehicles as mobile sensors (i.e. in-vehicle technologies or smart phones). Further, vehicle traffic routing system (VTRS) is a subset of ITS that is proposed to route the vehicles and reduce the travelers' commuting time by using the existing roads capacity. Generally, VTRS provides a path between two points (i.e. origin and destination) by considering different criteria. Besides, various adaptive traffic light systems (Gradinescu, Gorgorin, Diaconescu, Cristea, & Iftode, 2007; Maslekar, Boussedjra, Mouzna, & Labiod, 2011; Sanketh, Subbarao, & Jolapara, 2010; Tubaishat, Shang, & Shi, 2007; B. Zhou, Cao, Zeng, & Wu, 2010) operated by the aid of wireless communication among fixed RSUs and vehicles have been deployed and studied to monitor and manage traffic congestion conditions in the intersections. It should be mentioned that these systems are out of scope of this study, since the main focus and concern of this thesis is on VTRSs.

During three last decades, various VTRS technologies have been studied and developed around the world via utilizing different schemes. The first generation of VTRSs is in-car navigation systems that used only the distance metric without considering the traffic conditions in computing the shortest path to a destination. However, with advances and deployments of traffic surveillance infrastructure on the roads, the researchers have started to design applications that present the current vehicle traffic conditions to drivers and let them find better paths towards their destinations. However, these systems are not capable of providing accurate traffic information of all roads and consequently, cannot avoid congestion due to lack of adequate infrastructure for covering all the roads at once. The obtained traffic data along with road maps are used by VTRSs such as TomTom and Garmin, to find the shortest path or optimal path from source to destination. Unlike a large amount of researchers who focus on using static algorithms (e.g. Dijkstra (Dijkstra, 1959) and A* (Hart, Nilsson, and Raphael, 1968)) for finding shortest path such as Nazari, Meybodi, Salehigh, and Taghipour (2008), Noto & Sato (2000), Yue and Shao (2007) in VTRSs. Recently, researchers have focused on finding optimal paths by utilizing dynamic and meta-heuristic algorithms while considering various criteria (Boryczka & Bura, 2013). This trend happens due to the dynamic nature of vehicular environments which depend on both recurring (predictable) and non-recurring (unpredictable) congestion conditions and also because of multi-criteria nature of vehicle routing procedure. Recurring congestion refers to the regular traffic congestion which occurs due to limited capacity and inefficient traffic management. While, non-recurring congestion conditions which are irregular events which occur due to unpredictable events such as bad weather, car accident or breakdown and road construction. Moreover, multi-criteria shortest path problem (Ghoseiri & Nadjari, 2010; Martins, 1984) means that the distance is not the only objective of drivers and systems. Besides, congestion condition, number of traffic lights, number of roads' lanes, route safety, travel time and travel speed are some of the other objectives. Hansen (1980) proved that multi-criteria shortest path problem is a NP-problem.

These days, Google and Microsoft can predict vehicle congestion of road networks and its estimated duration by performing advanced statistical predictive analysis of traffic information. This traffic information is provided by some infrastructures (e.g. RSUs) to propose a traffic-aware shortest path for users and drivers. Therefore, their information is not only based on current traffic information but also on some other metrics such as weather and historical traffic information. However, it is worth noting that their systems are reactive and do not avoid vehicles congestion. This is because a same path is suggested to all of the users by these systems and similar to static algorithms, congestion is switched from one route to other one if a significant number of drivers utilize these systems. A more serious issue is that these systems do not consider non-recurring congestion efficiently in their routing procedure. These types of congestion include more than 50% of all vehicle congestion conditions (Coifman & Mallika, 2007).

Dynamic Traffic Assignment (DTA) can be counted as the second type of VTRSs that utilizes mathematical approaches and methods to model vehicle traffic dynamics and changes throughout the road networks. The main aim of DTA is to compute an optimal path for each driver in a way that none of the drivers can enhance his/her travel time by individually selecting an alternative path (Wardrop, 1952). Simulation based methods are used by most of the existing DTA tools to model the changes and dynamics of the road networks and compute drivers' route assignments iteratively. However, computation overhead is the main problem of DTA tools. A lot of iterations are required to achieve acceptable accuracy and convergence rate and each DTA iteration needs a sophisticated simulation procedure which leads to enormous computational overhead. Sensitivity, tractability for large road maps, ability of providing real-time routing guidance, convergence, accuracy of traffic changes, behavior in congestion condition and robustness to drivers who ignore the guidance are some other issues and drawbacks of DTA tools (Chiu et al., 2011).

In spite of significant advances of in-car navigation systems (e.g., TomTom and Garmin), web-based applications and services for route finding (e.g., Microsoft and Google Maps), and DTA tools, the traffic congestion problem has not been solved yet and its negative effects on drivers and the environment have still remained, especially in the bustling metropolises. The new advances in communication technology and vehicle industry lead to emersion of a new type of ad hoc networks, namely Vehicular Ad hoc Networks (VANETs). Most of the new vehicles are equipped with sensors, Global Positioning System (GPS) receivers and On-Board Units (OBUs) which can create vehicular networks. VANETs can be used to gather accurate real-time traffic information which can be utilized in VTRSs. VANETs is discussed extensively in Chapter 2.

In addition, swarm intelligence algorithms are newly emerged algorithms which simulate the behavior of different animals in the nature such as ants, bees, fishes and birds (Ahmed & Glasgow, 2012). These algorithms are able to produce fast, multi-criteria, low cost and robust solutions for various problems (Blum & Merkle, 2008; Panigrahi, Hiot, & Shi, 2010). Among these bio-inspired algorithms, using ant-based algorithms are reported as one of the best and promising approaches for vehicle traffic congestion control and management in many researches (S. Dhillon & P. Van Mieghem, 2007; J. Liu, Fang, & Liu, 2007; Bogdan Tatomir & Rothkrantz, 2004). However, the negative point of the ant-based algorithm is that it handles non-recurring congestion situations with some delay due to the stochastic feature of its searching approach and this causes many vehicles to unknowingly join in the congestion before the routing tables are updated. Ant-based algorithms are extensively discussed in the next chapter.

1.1. Statement of Problem

As mentioned before, vehicle traffic congestion has received a lot of attention from academic researchers due to its important effect on people’s daily life. However, it is not necessary to say that drivers not only desire the shortest path from source to destination but also desire safe journey with less travel time and congestion at the same time. It means that drivers want to be guided through the shortest path while considering other criteria such as traffic congestion, road width, risk of collision, and the number of intersections simultaneously in routing procedure. However, it does not mean that this route should be the shortest one, but, it should be optimal or quasi-optimal in terms of the mentioned preferences. Attention to this issue has increased during last decade due to the high number of vehicles and commutes, and VTRSs have been proposed as a solution. Nevertheless, there are still several important issues in this research area that have not been completely solved.

Initially, VTRSs found single shortest path in the static environments using static algorithms. Hence, dynamic and quick changes which are the main characteristics of vehicular environments in addition to the vehicle congestion are ignored by them. Therefore, static approaches are not suitable for vehicles routing as well as congestion avoiding (Wedde et al., 2007). Recently and because of these issues, routing mechanisms are modified to address dynamic changes of vehicular environments. In order to overcome these changes, new techniques and communication technologies such as camera, road wireless sensors and inductive loop detection are utilized to collect and monitor real-time traffic information. This information is used by routing mechanisms to find the optimal route for various origin and destination (OD) pairs. Figure 1.1 illustrates the difference between static and dynamic vehicle traffic routing. Distance is the only metric which is used for finding the shortest path in the static algorithms, while, dynamic algorithms (e.g. evolutionary, bio-inspired and local search algorithms) consider various metrics such as congestion, travel speed and travel time in their paths selection procedure.

(a) (b)

Figure 1.1: Comparison between proposed paths by (a) static and (b) dynamic algorithms for the same OD pair

Moreover, it is worth noting that half of the vehicle traffic congestion is related to non-recurring congestion conditions. However, this type of congestion is not considered in most of the existing VTRSs or its detection takes a long time via the existing methods (i.e., by using cameras and image processing methods). Even after identifying the street/road, in where the non-recurring congestion condition has happened, this road is just omitted (i.e. ignored) from path finding procedure until that road is released from non-recurring congestion condition. It should be mentioned that non-recurring congestion is not predictable and its detection takes a long time via the existing methods. Consequently, more vehicles are involved in the congestion before any re-routing occurs. One of the challenges is how to detect and consider the non-recurring congestion in a high dynamic vehicular environment by the aid of VTRSs. Moreover, traffic congestion occurs gradually; while, it takes a long time to recover from this condition. As a result, congestion avoidance is a better solution than recovering from it. In addition, drivers with the same origin and destination are routed through a same path by the existing VTRSs. Consequently, when a significant number of drivers utilize these systems (i.e., system usage rate is high), the congestion will be transferred from one route to another. Hence, how to make the system scalable to potentially high number of simultaneous and on-demand routing requests in terms of both computation and communication is another challenge.

In recent years, VTRSs based on swarm intelligence (SI) (James Kennedy, Kennedy, & Eberhart, 2001) algorithms, bio-inspired computation in general (Yang, Cui, Xiao, Gandomi, & Karamanoglu, 2013), have been widely used to overcome the vehicle traffic congestion problem (Kroon & Rothkrantz, 2003; Suson, 2010; Tonguz, 2011). SI is an emerging area in the field of optimization and researchers have developed various algorithms by modeling the behavior of different swarm of animals and insects such as ants, termites, bees, birds, fishes and bats (Yang et al., 2013). Among these bio-inspired algorithms, using ant-based algorithms are reported as one of the best and promising approaches for vehicle traffic congestion control and management in many researches (Dhillon & Van Mieghem, 2007; Liu et al., 2007; Tatomir & Rothkrantz, 2004). Ant-based approaches outperforms the other bio-inspired algorithms in both processing time and adaptability which are the most important factors in designing vehicle routing and congestion avoidance approaches (Elbeltagi, Hegazy, & Grierson, 2005; Wedde & Senge, 2013). However, ant-based algorithms suffer from scalability and delay while updating their routing tables in order to consider congestion conditions. These drawbacks along with the mentioned shortcomings for the existing approaches (i.e. ignoring non-recurring congestion and routing vehicles with the same OD pair via single path) are the main concerns of this thesis. This thesis addresses the necessity of existence of a proper VTRS which routes the vehicles and avoids traffic congestion simultaneously, instead of recovering from congestion after its occurrence. It means that the optimal paths which are not necessarily the shortest paths, but the fastest paths in time, called least congested shortest paths, are proposed to drivers. In order to considering both recurring and non-recurring congestion conditions, different metrics such as travel time, speed, distance, density are considered in optimal path finding procedure.

1.2. Research Objectives

In general terms, the aim of this thesis is providing robust vehicle traffic routing framework that will avoid congestion by providing least congested shortest paths as alternative routes for drivers by taking advantage of VANET and ant-based algorithm. Robustness is related to the adaptability and flexibility of the system against various conditions. The objectives which are defined to achieve the main aim are as follows:

- To present a classification for the existing research trends within the area of vehicle traffic routing and congestion control,
- To provide a taxonomy and statistical overview of ant-based approaches, which are used in vehicle traffic routing systems,

- To design a scalable vehicle traffic congestion avoidance framework using ant-based algorithm,
- To develop the proposed framework in simulation environment using network simulator, mobility generator and VANET simulator,
- To evaluate and analyze the proposed framework with different set of scenarios and evaluation metrics.

1.3. Scope of Study

Designing an efficient VTRS is a multifaceted task as depicted in Figure 1.2. This figure represents a comprehensive overview of the VTRS essential prerequisites. For instance, VTRS requires a system architecture, path selection and map preparation methods. The words written in red color indicate the methods which are used in our proposed approach. In other words, they represent the scope of this study. For example, although there are various methods for gathering real-time traffic data such as RFID, VANET, loop detector and cameras, VANET is used for this purpose in our approach and the other methods are out of scope of this study. One of VTRSs main complexities is related to the interaction of various criteria such as distance, traffic load, road width, risk of collision, number of intersections, weather condition, dynamic changes of vehicular environments and special events in path selection procedure. The other point that should be mentioned regarding Figure 1.2 is that traffic light optimization, vehicle's type consideration and speed limit dissemination can be considered as an extension of VTRS in order to alleviate the traffic congestion problem. These methods are not discussed and considered in this study.

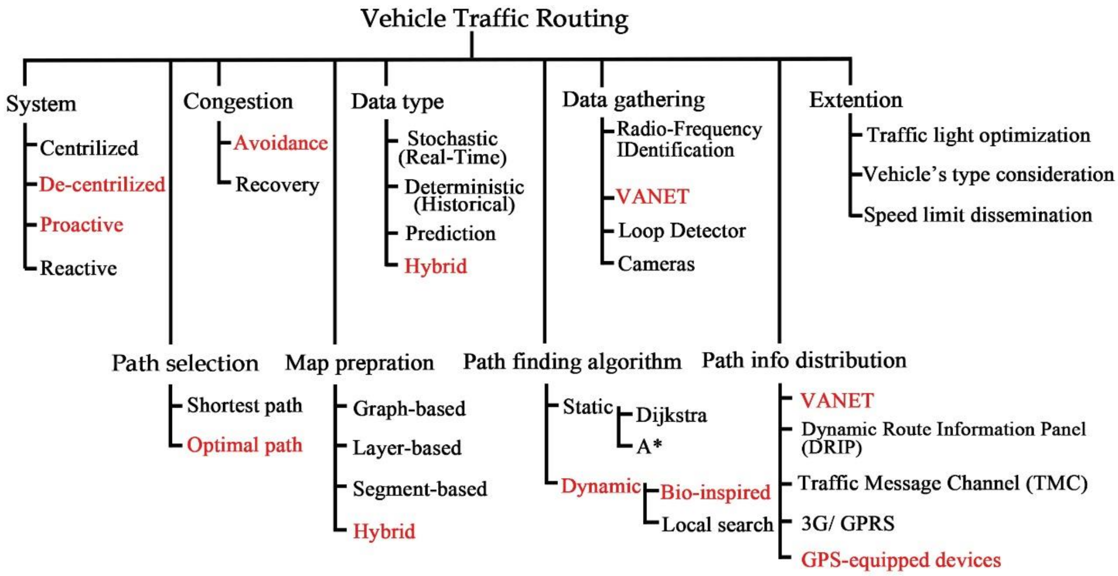

Figure 1.2. Essential prerequisite for VTRSs

In addition, some other limitations of the scope of this thesis are as follows:

- Firstly, motorized vehicles such as cars, vans and trucks are considered in routing process. Motorbikes, bicycles and pedestrians are not considered and are beyond the scope of this thesis.
- Although vehicle traffic routing systems include man-machine interactions (i.e., user interface of system), this issue is not the concern of this thesis. In other words, quality of experience or user experience is out of the scope of this thesis.
- Moreover, vehicle routing problem (Golden, Raghavan, & Wasil, 2008; Toth & Vigo, 2001) is selection of a set of optimal paths for distributing some products between desired customers by a fleet of vehicles. This problem is out of scope of this thesis and is completely different from vehicle traffic routing and congestion problem which is the main focus of this thesis.

1.4. Motivations and Significance of Study

Vehicle traffic congestion is a problem which leads to lots of negative side effects for people, governments and the environment in different ways. From people's point of view, congestion imposes delays, financial loss, accident and change of travel behavior (i.e., they have to depart earlier or choose other types of transportations such as bus or electric railway). Based on investigations, congestion can lead to anger and stress responses, and provokes rage of frustrated motorists. It also affects the environment by increasing fuel consumption which leads to high level of air pollution and CO_2 and greenhouse gases emissions.

On the other hand, congestion has direct effects on governments mainly in economic and environmental issues, since they have to expand roadways infrastructures by building new highways, installing new traffic lights and traffic monitoring systems. In addition, governments need to spend more money for buying or producing extra fuels due to high fuel consumption as a consequence of vehicle traffic congestion. Moreover, they have to dedicate some budgets for training and advertising the ways of reducing congestion.

From environmental aspects, vehicles are one of the largest sources of air pollution in the world. Surprisingly, vehicles emit more pollution when moving at slower speed. Therefore, vehicles emit more CO_2 and greenhouse gases at traffic jams. CO_2 contributes to climate change by insulating more heat from the sun. Ozone can impair lung function, especially in the elderly, children and adults with asthma, with a higher number of sufferers resulting in high-traffic urban areas. Besides, vehicles make excessive noise when moving together which leads to noise pollution in big cities. Moreover, the environment must be changed in order to expand the roadways infrastructures. These changes can lead to destruction of forests and natural habitat which are the main sources of oxygen for human beings.

Considering the enormous costs and harmful effects of vehicle congestion on people, governments and the environment, finding effective solutions is necessary. Therefore, reducing or avoiding vehicle congestion not only solves one of the main transportation concerns but also decreases a lot of its harmful effects on human life. In addition, the recent advances in sensing, communication and computing technologies (e.g., VANETs and bio-inspired algorithms) encourage us to involve and utilize them in VTRS in order to mitigate the vehicle traffic congestion problem. The work presented in this thesis can contribute to the solution of congestion in the following ways:

- Dynamic re-routing and providing alternative routes based on real-time and predicted traffic information can reduce the impact of non-recurring congestion on traffic jams.
- The waste of money and time of travelers will be reduced by providing least congested shortest paths which decreases and increases average travel time and average travel speed, respectively.
- Adaptive vehicle traffic routing system, which proposes less congested shortest paths for drivers, will reduce fuel consumption, harmful gases (e.g. CO_2 and greenhouse gases) emission and also the stress and anger level of drivers feeling.

1.5. Thesis organization

A general overview of the topics discussed in this thesis is presented in this section as follows:

The problem statement, relevant research objectives, scope of the study along with our motivations and research significant were represented in introduction section.

Chapter 2: This chapter provides the required background knowledge for the present study. It explains what VANETs are and its related standards, characteristics as well as applications. Various communication types that are exist and used in vehicular networks are also discussed in this chapter. An overview of bio-inspired algorithms and their various types is provided in this chapter. Particularly, ant-based algorithm along with its concept, types and applications are represented in Chapter 2, since its concept is the cornerstone of our proposed approach.

Chapter 3: It includes the related work to the thesis topic. In the first part, the most important characteristics of VTRSs are discussed in detail. The second part of this chapter is concerned with static routing mechanisms that are used in VTRSs. An overview of dynamic routing mechanisms with more focus on bio-inspired algorithms is discussed in the third part of Chapter 3. In addition, a critical review on ant-based VTRSs along with our proposed taxonomy for these approaches is represented in a subsection of this chapter. Our proposed taxonomy includes three main classes, namely, Ant Colony Parameters (ACPs), Ant Colony Prediction (ACPre), and Ant Colony Segmentation (ACS). Chapter 3 is concluded with a statistical discussion on ant-based VTRSs.

Chapter 4: This chapter provides the detailed aspects of developed framework, called Ant-based Vehicle Congestion Avoidance Framework (AVCAF). The general steps of our research methodology along with used research method are represented in the first section of this chapter. In the second section, the challenges that should be considered for designing an ant-based VTRS framework are disused in details. A description of our developed AVCAF framework along with its phases, namely initialization, optimal path finding and optimal path suggestion are discussed in third part of Chapter 4.

Chapter 5: This chapter is dedicated to the explanation of the implementation details regarding AVCAF framework in simulation environment. After providing a comprehensive overview of existing simulation software and tools that are developed for VANET based approaches and applications, Network Simulator-2 (NS-2), Simulation of Urban MObility (SUMO) and Traffic and Network Simulator (TraNS) are selected as network simulator, mobility generator and VANET simulator, respectively, to implement AVCAF. The third part of this chapter is devoted to the presentation of the implementation procedure and details as well as simulation setups which are used for developing our approach (i.e. AVCAF framework) by taking advantage of the mentioned simulation software.

Chapter 6: This chapter goes through explanation of the obtained results from running the implementations and their explanations in order to evaluate the effectiveness of our proposed approach. In the first part, finding the proper value for AVCAF parameters via examining various values for them through the simulation tool is discussed. In addition, some approaches are selected as benchmark system in order to compare with our proposed approach. Various scenarios along with different evaluation metrics are defined and designed to ensure that our main aim is achieved by the proposed approach. The last part of this chapter includes the obtained results and their discussions for evaluated approaches considering various designed scenarios and evaluation metrics.

Chapter 7: This chapter concludes the thesis by providing an overview of the problem statement, research purpose, reached goals, objectives and findings. Moreover, all of the main points and obtained results are represented. This chapter concludes by discussing open issues and challenges in VTRSs and proposing some directions as a future work section.

CHAPTER 2: BACKGROUND

The architecture of VTRS can be divided into two categories, namely centralized and decentralized (Suson, 2010). In a decentralized architecture the route finding and computation take place for each vehicle individually using on board processor and storage of that vehicle. It is ideal if vehicles receive traffic information through wireless communications (e.g. vehicle-to-vehicle or vehicle-to-infrastructure) and include road map and GPS. While, in a centralized architecture the route finding and computation takes place by a central server in response to requests from drivers. In this architecture, central server has access to historical or real-time traffic information database and compute routing algorithm based on this information. Centralized routing architecture provides better reliability and visibility for vehicle routing and routing map, respectively. Easier results analysis is another advantage of central architecture. While, it is vulnerable in case the system's server breaks down and also suffers from scalability issues. Although decentralized approaches do not suffer from these drawbacks and reduce data process time and required storage spaces, currently, they cannot be implemented due to lack of infrastructures (i.e. all vehicles are not equipped with on-board units and wireless transceivers) and most of the algorithms should be modified and some extra communications are needed in order to use decentralized architectures for VTRSs. More information about centralized or decentralized architecture and their pros and cons can be obtained in study by Suson (2010). However, distributed centralized architecture can solve most of the mentioned drawbacks for these two types of architectures. Distributed centralized architecture is discussed in more details in Chapter 4.

It is worth noting that in all VTRS architectures, some data should be gathered or provided and an algorithm is used to find the optimal paths for vehicles based on this data. The title of this thesis and a brief discussion in introduction section indicate that our proposed AVCAF framework uses VANET and a novel version of Ant Colony Optimization (ACO) for gathering real-time traffic data and vehicle routing, respectively. Hence, these two subjects are discussed extensively in the following sections.

2.1. Vehicular Ad hoc Networks

This section discusses about VANET technology which belongs to wireless communication networks area. The increasing demand for wireless communication and the need for new wireless devices have led to research on self-organizing or self-healing networks without the interference of centralized or pre-established infrastructure/authority. Any network, in which a centralized or pre-established infrastructure is absent, is called an Ad hoc network. There are different classifications of wireless ad hoc networks which are based on their application, such as Mobile Ad Hoc Networks (MANETs), Wireless Mesh Networks, Wireless Sensor Networks and VANETs. VANET is the subclass of MANET, in which vehicles act as nodes (Zhu, Niyato, Wang, Hossain, & In Kim, 2011). Unlike MANET, vehicles move on predefined roads and have to follow traffic signs and signals (Taleb et al., 2007). Differences between VANETs and MANETs are summarized in Table 2.1.

Table 2.1. MANET vs. VANET

Parameters	**MANET**	**VANET**
Cost of production	Cheap	Expensive
Change of topology	Slow	Frequently and very fast
Mobility	Low	High
Node density	Sparse	Dense and frequently variable
Bandwidth	100kps	1000kps
Range	Up to 100m	Up to 500m
Node lifetime	Depends on power resource	Depends on lifetime of vehicle
Multi-hop routing	Available	Weakly available
Reliability	Medium	High
Moving pattern of nodes	Random	Regular
Addressing scheme	Attribute-based	Location-based
Position acquisition	Using ultrasonic	Using GPS, RADAR

VANET is one of the most influencing areas for the improvement of ITS in order to provide safety and non-safety applications for the roads' users. The frequencies for VANET wireless communication were allocated by the Federal Communication Commission (FCC). Subsequently, in 2003, FCC established the Dedicated Short Range Communications (DSRC) Service, which is a communication service operating in public and private safety at a frequency range from 5.850 to 5.925GHz (Jiang & Delgrossi, 2008). Due to high vehicular mobility, faster topological changes, requirements of high reliability and low latency for safety applications, the original 802.11 protocol is not suitable and adaptable for VANETs (Darus & Bakar, 2011). Wireless Access in Vehicular Environments (WAVE) standard, or IEEE 802.11p, is the subject that IEEE worked on, with the purpose of providing DSRC for VANET communication. Figure 2.1 displays the multi-channel system created in DSRC. The DSRC spectrum is divided into seven channels by the FCC so that each of them has 10 MHz bandwidth. Six of them were identified as service channels, and one of them is identified as the control channel, as shown in Figure 2.1. The control channel is used for safety messages, while service channels are used for non-safety as well as WAVE-mode messages or services (Amadeo, Campolo, Molinaro, & Ruggeri, 2009; Mak, Laberteaux, Sengupta, & Ergen, 2009).

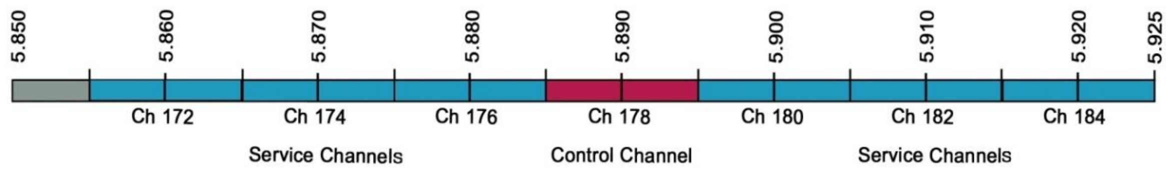

Figure 2.1. DSRC Channels

VANET technology creates a mobile network by using moving vehicles as nodes in a network. Every node within VANET acts as sender, receiver and network router, as each node communicates through other intermediate nodes lying within its own transmission range and creates a network with a wide range. Although VANET is a self-organizing network and it does not rely on any fixed network infrastructure, some fixed nodes act as RSUs or base stations to facilitate the vehicular networks communications and services. The communication equipment (i.e. On-Board Unit (OBU)) used in these vehicles and RSUs enable them to send and receive messages between each other via Vehicle-to-Vehicle (V2V)/inter-vehicle and inter-roadside communications, and also to send and receive messages to/from network infrastructures on roadside via Vehicle-to-Infrastructure (V2I) or Vehicle-to-Roadside (V2R) communication. Vehicles accurate position is required in most of the applications and protocols. Hence, some positioning hardware such as GPS and Differential Global Positioning System are also embedded in vehicles. The operation of vehicular networks is currently based on the exchange of two primary types of messages. On one hand, Cooperative Awareness Messages, also known as beacons, are broadcasted periodically by all nodes (vehicles) on the control channel, to provide and receive status information about the presence, geographical position and movement of neighboring nodes, and service announcements to/from those nodes. On the other hand, event-driven emergency messages are transmitted when an abnormal or dangerous situation is detected, in order to inform surrounding nodes about it. Figure 2.2 shows the overall working structure of VANET.

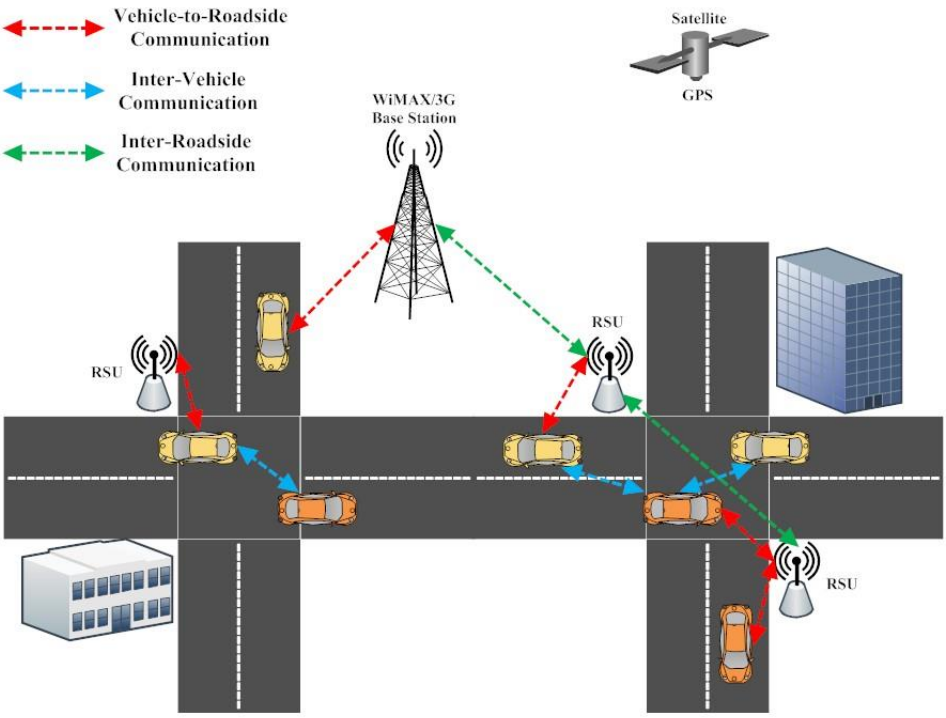

Figure 2.2. Overall VANET scenario

Although the main target of VANET is to increase safety of road users and passengers, drivers and passengers comfort, infotainment and entertainment are also provided and considered in VANET by researchers. The excitement surrounding vehicular networking is not only due to the applications or their potential benefits but also due to its unique characteristics, challenges and requirements. Hence, VANET applications and requirements and challenges are briefly overviewed in the following sub-sections.

2.1.1. VANET Applications

This section discusses main applications of vehicular networks and their examples. Although drivers and passengers safety is the main concern of VANET, their comfort and entertainment can be also considered and achieved by VANET. Hence, VANET applications can be classified into three classes, namely road safety applications, traffic and transport efficiency and management applications and entertainment/infotainment applications (Hartenstein & Laberteaux, 2008; Karagiannis et al., 2011). Each of these classes is discussed in more details as follows:

1) Road safety applications: The applications are primarily developed to reduce the probability of accidents and the loss of life of road users (e.g. drivers and passengers) (Sichitiu & Kihl, 2008). A significant percentage of accidents that occur every year in all parts of the world are associated with intersection, head, rear-end and lateral vehicle collisions. VANET assists vehicle drivers to communicate with each other and to provide information such as vehicle position, speed, direction, intersection position and vehicle distance from it, in order to avoid any critical situation (e.g. accidents or collisions) through V2V and V2I communications. These communications and information can be used to locate and to predict hazardous locations on the roads. Lane change assistance, intersection collision warning, emergency electronic brake lights, collision risk warning and pre-crash sensing/warning are some examples of this class of VANET applications.

2) Traffic and transport efficiency and management applications: This type of VANET applications is developed to enhance the traffic flow, congestion and coordination as well as traffic monitoring. These applications provide real-time traffic information for drivers and passengers which can be used for road map updating, vehicle routing or vehicle traffic congestion problem mitigation. Time and location are two important metrics in these applications due to dynamic and quick changes of vehicular environments. Speed management and co-operative navigation are two well-known groups of this class. In the former case, the application helps drivers to regulate and control the vehicle speed to prevent unnecessary stops and to smooth driving. Variable speed limit notification and green light optimal speed advisory are two examples of first group. In the latter case, the applications are utilized to enhance the vehicle traffic efficiency by routing and navigating vehicles via cooperation among vehicles and RSUs. Cooperative adaptive cruise control, VTRS and platooning are some examples of this group.
3) Entertainment/Infotainment applications: As the name of these applications indicates, they focus on providing entertainment for road users. These applications, similar to traffic and transport efficiency and management applications, can be divided into two groups, namely cooperative local services and global Internet services. First group’s applications concentrate on infotainment that can be achieved locally such as point-of-interest notification, local e-commerce and media downloading. Service announcements, real-time video relay and remote vehicle personalization/diagnostics are some examples of the first group’s applications. However, the infotainment data that can be achieved from global Internet services are the main concern of second group’s applications. Communities services (e.g. insurance and financial services, fleet and parking zone management) and ITS station life cycle services (e.g. software and data updates) are two well-known examples of second group’s applications.

2.1.2. VANET Requirements

As discussed in previous subsection, VANET provides wide range of applications for road users. However, there are many requirements in VANET that are needed to be considered in order to provide reliable services and applications. Therefore, this section discusses the main requirements of VANET applications. According to a study by Karagiannis et al. (2011), these requirements are classified into seven categories as follow:

1) Strategic requirements: These are related to vehicular network penetration threshold or deployment level and strategies defined by governments and commissions.
2) Economical requirements: These are related to financial issues such as required devices cost, ongoing cost and time for completing the application and return of the invested cost.
3) System capabilities requirements: Radio communication, network communication, vehicle positioning and communication security are the most important system capabilities requirements.
4) System performance requirements: Communication performance (e.g. delay, delivery ratio, and throughput), vehicle positioning accuracy, reliability and dependability of system (e.g. coverage area, bit error rate) and security operations (authentication, authorization and verification) are the most notable system performance requirements of VANET applications.
5) Organizational requirements: These organizational activities related to deployment of vehicular networks including consistent naming repository and address directory, ensuring the interoperability between different ITSs, considering security requirements.
6) Legal requirements: Supporting road users' privacy, their liability/responsibility and lawful interception are the main legal requirements of VANET applications.

7) Standardization and certification requirements: System and ITS station standardization, service conformity and system interoperability testing, and system risk management should be supported in order to obtain standardization and certification requirements.

2.1.3. VANET Challenges

Several types of VANET applications and their requirements are discussed in sections 2.1.1 and 2.1.2, respectively. Wide range of requirements for VANET applications leads to a number of research challenges in VANET as follows:

1) *Addressing:* Some VANET applications require that the addresses should be linked to the physical or geographical position of a vehicle. This addressing issue is challenging due to high mobility and quick changes of vehicles.
2) *Risk analysis and management:* This challenge is related to identifying and controlling the threats and potential attacks in various types of VANET communications (i.e. V2X). Although some solutions are proposed for managing these attacks, attacker behavior models have not been investigated in details.
3) *Data-centric trust and verification:* Data trustiness is more important and useful than the sender node trustiness for most of the VANET applications. This challenge is related to providing security means for VANET applications to ensure that the received information is trustable and its integrity can be verified in order to protect the vehicular network from the in-transit traffic tampering and impersonation security threats and attacks (Raya, Papadimitratos, & Hubaux, 2006). Although public key cryptography can be considered as a solution for this challenge, the introduced overhead via this solution is its main drawback (Zhou & Haas, 1999).
4) *Anonymity, privacy and liability:* The source, which can be either a vehicle or any other network entity, of received information should be trustable in VANET application. Simultaneously, drivers' privacy is a basic right, which is protected by

law in most of the countries, and can be obtained via nameless vehicle identities. The main challenge is associated with the tradeoff between authentication, privacy and liability, when the network has to reveal the sent information and also its source to some authorized entities by government.

5) *Secure localization:* This is a Denial of Service resilience strategy which protects the VANET against attackers that are intentionally willing to find the location and position of the vehicles.

6) *Delay constraints:* Data packets sent via VANET applications have time and location significance. Designing vehicular communication protocols with good delay performance is another VANET challenges due to vehicles' velocity, mobility, unreliable connectivity and consequently, quick topological changes.

7) *Prioritization and congestion control:* Safety and traffic efficiency messages have higher priority and should be delivered faster that other messages. Most of the existing researches concentrated on finding a way to provide highest priority for safety and emergency messages. However, the channel is congested due to its limit bandwidth and also broadcast nature of emergency messages. In-network aggregation protocols are proposed for enhancing communication efficiency in VANET by combining and aggregating information from multiple sources during packet routing procedure. A comprehensive overview of in-network aggregation mechanisms for VANET can be found in the study by Dietzel, Petit, Kargl, and Scheuermann (2014).

8) *Reliability and cross-layering between transport and networking layers:* wireless communication, which is very sensitive and unreliable due to frequent disconnectivity, is the cornerstone of vehicular networks. Hence, providing reliable transport service on top of the unreliable network is one of the major challenges of

VANET. Designing cross-layer protocol, which span between transport and routing layers, can be a solution for this challenge.

Comprehensive and detailed discussion on solutions for the mentioned VANET challenges can be found in the study by Karagiannis et al. (2011).

2.2. Bio-inspired algorithms

Bio-inspired and SI algorithms have attracted the attention of many researchers and scientists for the last two decades. These algorithms are an emerging area in the field of optimization and researchers have developed various algorithms by modeling the behaviors of different swarm of animals and insects such as ants, bees, birds, firefly, Cuckoo and bats (Yang et al., 2013). The two main reasons for these algorithms popularity are as follows: 1) these are very flexible and versatile, and 2) These algorithms are very efficient in producing fast, multi-criteria, low cost and robust solutions for dynamic and nonlinear problems (Blum & Li, 2008; Panigrahi, Shi, & Lim, 2011; Yang et al., 2013). Bio-inspired computation has pervaded into almost all areas of sciences, engineering, and industries, and it is one of the most active and well-known research areas due to wide range of its applicability. Wide range of bio-inspired algorithms is proposed by researchers in recent decades. An overview of various types of bio-inspired algorithms is provided in the following subsection. These algorithms are including but not limited to ant, bee, bat, particle swarm optimization (PSO), firefly, cuckoo and genetic algorithms.

2.2.1. Various types of bio-inspired algorithms

1) Ant algorithm

Ant-based algorithms mimic the cooperative behavior of real ants in finding food resources. By investigating ant behavior, researchers have recognized that each ant randomly explores its surrounding area to find food resources. Upon finding food, an ant uses a chemical material known as pheromone to inform other ants about the food source. Individual ants can perform tasks independently while collaborating with other ants to solve a problem (i.e., finding food sources). Since ant algorithm is the cornerstone of our proposed approach, its steps, types and applications are discussed in more details in Section 2.2.2.

2) Bee algorithm

Similar to ant algorithm, bee algorithm is inspired from foraging behavior of real honeybees in nature. Waggle dance, nectar maximization and polarization are the most interesting characteristics of real honeybees which are used to simulate the allocation of the forager bees around flower lots and consequently, various search areas in the search region. Different types of bee algorithm are proposed by researchers as follows: in most of the existing bee algorithms, called honeybee-based algorithms, different forager bees are allocated to different nectar resources in order to maximize the obtained nectar (Karaboga, 2005; Nakrani & Tovey, 2004; Pham et al., 2006). Yang (2005) proposed virtual bee algorithm in which pheromone intensity is directly linked with the objective function. Two types of bees are utilized in honeybee-based algorithms, while, artificial bee colony algorithm divides bees into three types, namely employers, onlookers and scouts. More information about bee algorithms and their applications can be found in the study by Yang (2010a) and Parpinelli and Lopes (2011).

3) Bat algorithm

Echolocation behavior of micro-bats leads to development of bat algorithm by Yang (2010b). Echolocation is a type of sonar which is utilized by micro-bats to find foods, avoid obstacles and locate their roosting crevices in the dark. Micro-bats emit loud sound pulses and listen to their echoes that return back from the surrounding objects. The specifications of these pulses related to bats species and their hunting strategies. Bat algorithm is based on three rules: 1) echolocation is used by all bats to sense distance, and they can differentiate foods from background barriers. 2) A bat prowl randomly with speed of v_i, at location of x_i with a fixed frequency range [f_{min}, f_{max}]. In addition, emission rate, r, and plus loudness, A_0, are tuned based on the position of targeted prey. 3) Although the plus loudness can vary in many ways, it should vary from a large, A_0, to a minimum constant value, A_{min}. An extension of bat algorithm, called multi-objective bat algorithm is developed by Yang (2011a) and experimental results by Yang and Gandomi (2012) represent its efficiency in solving multi-objective problems.

4) Particle swarm optimization

Kennedy and Eberhart (1995) proposed PSO based on swarm behavior of bird flocking and fish schooling. The space of an objective function is explored by this algorithm through adjusting the particles, which are the direction flow of each agent, as like piecewise paths represented by location vectors in an approximately random manner. The motion of a swarming particle includes two main ingredients, namely stochastic and deterministic. It means that every particle has tendency to move toward the existing global best and its own best position in its history, while, it wants to move randomly, simultaneously. Accelerated PSO is another type of PSO which is developed by Yang, Deb, and Fong (2011) in order to overcome business optimization problems.

5) Firefly algorithm

Firefly algorithm was proposed by Yang (2009) based on the flashing pattern and real fireflies' behavior. This algorithm follows three of the following rules: 1) Fireflies are unisexual and they will be attracted to each other regardless of their sex. 2) Attractiveness has a direct relationship with brightness, while, it has inverse relationship with distance. For any two flashing fireflies, the less brighter one will move toward the more brighter one. They will move randomly, if there is not any brighter firefly in the surrounding area. 3) The firefly brightness is related to the landscape of the objective function. Although NP-hard scheduling problems can be solved by discrete version of firefly algorithm, it can be used for solving wide range of test problems, containing multi-objective load dispatch problems (Apostolopoulos & Vlachos, 2010). A new firefly algorithm, called chaos-enhanced firefly algorithm, is proposed by Yang (2011b) for automatic parameter setting.

6) Cuckoo search

Another bio-inspired algorithm which is developed by Yang and Deb (2009) is named cuckoo search. This new algorithm is based on the brood parasitism of some cuckoo species. Based on the existing literature and experiments, cuckoo search outperforms PSO and GA in solving structural optimization problems (Gandomi, Yang, & Alavi, 2013). Aggressive reproduction strategy is the most important and notable characteristic of cuckoos. This algorithm follows three of the following rules: 1) One egg is laid by each cuckoo at a specific time and is located in a randomly chosen nest. 2) The nests with high quality eggs are called best nests and will be preserved for the next generations. 3) Pre-defined and fixed number of host nests are available, and the probability of recognizing an egg which is laid by another cuckoo and is not belong to the host nest by host bird is $P_a \in [0, 1]$. If this is the case, the alien egg will be removed from the nest by host bird or the host bird leaves the next and creates a new nest. Engineering optimization problems can be solved by cuckoo search in an efficient manner (Gandomi et al., 2013).

7) Genetic Algorithm

Genetic algorithm (GA) (Goldberg, 1989) is adaptive and robust search technique inspired by natural selection and genetics in biology. GA is population based algorithm that encodes a potential solution in a form of string data structure, namely chromosome. Each chromosome (i.e. solution) consists of a set of elements, namely genes, which hold a set of values for optimization variables. Fitness or objective function is used to determine the suitability of each solution for desired problem. Selection, crossover and mutation are three basic used operations of GA. The selection operation chooses nonoverlapping random sets of two chromosomes from initial population and then selects the chromosome with higher fitness value from each set to survive in the next iteration (i.e. generation). Crossover operation includes two steps: 1) determining a gene in a random way as a crossover point from two parent chromosomes, 2) mutual exchange of chromosomes by considering crossover point and generating new off-springs. Crossover used to improve the fitness value of solutions. The last but not the least, a series of chromosome’s genes is changed after determining mutation point via mutation operation in order to increase population diversity and avoid falling into a local optimal solution. However, loops and short circuits generated by mutation operation should be prevented through a control mechanism (Xu & Ke, 2008).

It is worth noting that SI algorithms are a major subset of metaheuristic algorithms. There are some common challenges and open issues among SI algorithms as follows: theoretical convergence speed analysis, proper classification, taxonomy and terminology, tuning algorithm-dependent parameters and examining existing algorithms for large-scale problems.

2.2.2. Ant-based algorithm

How do real ants communicate with each other in finding food sources and accumulate food in their nest using the shortest path, considering the fact that they are blind insects? This question has attracted the attention of many researchers and scientists for many years. The answer is that the ants release a chemical liquid, called pheromone, on their traversed paths based on the quality of the found food resource while moving from their nest to the food source and vice versa. This pheromone trail helps other ants to find the food resources by sniffing the pheromone. The pheromone intensity decreases (pheromone evaporation) over time in order to increase the probability of finding new paths. This issue that the ants find and use the shortest path between their nest and the food source has been proved both mathematically and experimentally by researchers.

Mathematical proof is done by Shah, Bhaya, Kothari, and Chandra (2013), while double bridge experiment is used for experimental proof. Double bridge experiment is depicted in Figure 2.3 in which there are two different bridge (i.e. paths) between source (i.e. nest) and destination (i.e. food source).

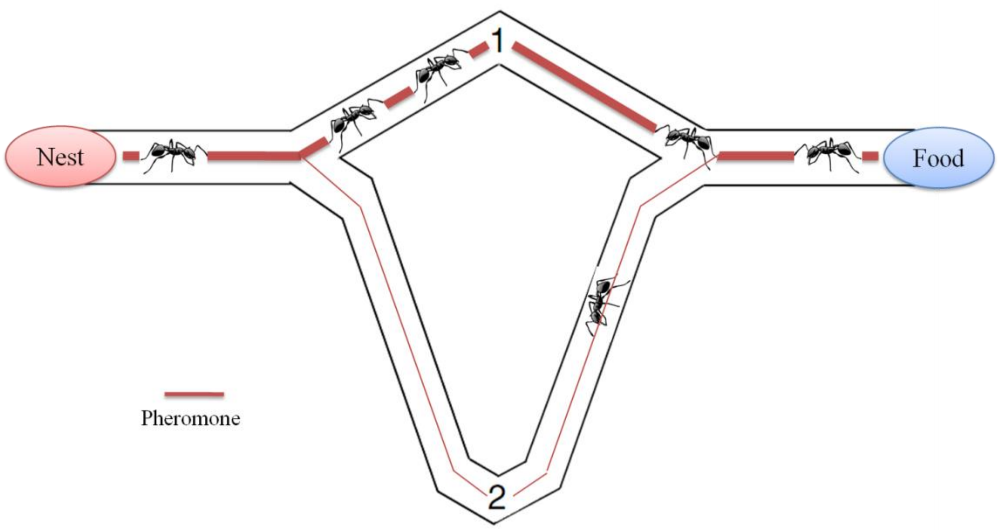

Figure 2.3. Double bridge experiment (Marco Dorigo & Birattari, 2010)

Length is the main difference between these two bridges where bridge 1 is shorter than bridge 2. Initially, the ants explore the surrounding environment for finding food sources by performing a random selection between bridges 1 or 2. However, the ants which select bridge 1 arrive faster to the food source and come back to the nest earlier that the ants which choose bridge 2 for food exploration. This is because bridge 1 is shorter than bridge 2. In this way, more ants will be attracted to bridge 1 due to pheromone existence on it. Therefore, the pheromone intensity will be increased on this bridge over the time, while, it will be reduced on bridge 2 due to pheromone evaporation. As a result, ant-based algorithm is an efficient way to find shortest path between two desired points.

This phenomenon forms the infrastructure of an Ant System (AS) and ACO algorithms proposed by Marco Dorigo (1992) and Marco Dorigo, Caro, and Gambardella (1999) to simulate real ant behaviors by using artificial ants.

Most of the characteristics of real ants are mimicked by artificial ants in order to simulate the behavior of an ant colony for the solution of optimization and distributed control problems. The most common characteristics of real and artificial ants are discussed as follows:

- *Pheromone trail based stigmergy communication:* Pheromone trails assistants to find the shortest path from nest to food source. Stigmergy communication is a self-organizing behavior of ants which is required to interact with each ant (Theraulaz & Bonabeau, 1999). This communication occurs in an indirect manner which means that an ant alters its surrounding environment by laying pheromone on its traversed paths and the other ants respond to this modification at a later time (Bonabeau, Dorigo, & Theraulaz, 1999). Stigmergy can be transferred to artificial ants by assigning numerical information to the problem space variables and by giving local access to these variables to the artificial ants (Marco Dorigo et al., 1999).
- *Implicit shortest path finding:* An implicit shortest path finding happens by reinforcement on the shortest path for both real and artificial ants. More pheromone is laid on the shortest paths because they are traversed more faster than longer paths (Di Caro, 2004).
- *Concurrent and independent iterations:* The artificial individual ant, similar to the real one, is able to find a path from nest to food resource, but it is not the only ant that does this action. The other ants do the same task concurrently and independently in order to converge to the optimal path in a short time (Marco Dorigo & Birattari, 2010).
- *Discrete world:* Artificial ants, unlike real ants, live in a discrete world which means that their actions are transitions from one discrete condition to another (Marco Dorigo et al., 1999).

- *Synchronized vs. desynchronized system:* Artificial ants move from their nest to the food source and vice versa in each iteration. Therefore, they move in a synchronized way unlike real ants which move in a desynchronized pattern (Blum, 2005).
- *Memory:* The artificial ant utilizes an embedded memory to store the traversed path information. It is used for building and evaluating possible solutions, for back tracking from the destination to the source and also for updating the pheromone value on the found path. In comparison, real ants do not have memory but use their sensing capability for this purpose.
- *Pheromone evaporation strategy*: Pheromone evaporation happens very slowly in nature and its rate is constant (Deneubourg, Aron, Goss, & Pasteels, 1990). This mechanism and its evaporation rate vary from one problem to another in the simulation environment for artificial ants.
- *Extra capabilities:* Artificial ants use extra capabilities to increase the efficiency of the whole system that can be augmented with capabilities such as future prediction and local optimization, while these capabilities cannot be found in real ants.

2.2.2.1. Ant Colony Optimization

ACO (Marco Dorigo et al., 1999; Marco Dorigo & Stützle, 2003) studies artificial systems which take inspiration from the real ants behaviors in nature in order to solve optimization problems. ACO algorithms follow four main steps as follows:

1) *Problem environment depiction*: artificial ants move from one discrete state to another. Therefore, they can solve discrete problems (LaValle, 2006) which can be depicted as graph with *N* nodes and *L* links.
2) *Initialization*: in this step, a number of artificial ants (*Na*) are located on the nodes (sources) and a specific value (weight) is assigned to each link of the problem graph. The re-generation period of ants (Υ), which means the time interval between two consecutive ant colonies generation is started periodically at a pre-defined time

interval for finding new paths. *Na* and Υ are obtained using experiments or trial and error. Moreover, physical distance, random number, queue length or a number obtained by mathematical formula can be used as initial weight of links. The node transition rule is defined and used for next node selection. The probability of choosing *j* as a next node from *i* by ant *k* is calculated by Equation 2.1.

$$P_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{h \notin tabu_k} (\tau_{ih})^{\alpha}(\eta_{ih})^{\beta}} & \text{if } j \notin tabu_k, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Intensities α and β are the relative importance that can be used to stress the importance of pheromone intensity τ_{ij} and η_{ij} route cost. $tabu_k$ is the set of visited nodes by ant *k*.

3) *Pheromone update*: ants start to move from source to destination using the node transition rule (Equation 2.1) and store visited nodes in their memory. Whenever an ant reaches to its destination, it backtracks to its origin/source node using its memory and updates the links' pheromone value on its return path using the pheromone update rule. Two concepts are embedded and considered in this rule. On one hand, the pheromone value of the links, which are not traversed by the ant, should be decreased in order to reduce their selection probability by the other ants. This issue is called pheromone evaporation and should be considered in the pheromone update rule. On the other hand, the pheromone value of the links, which are traversed by the ant, should be increased in order to enhance their selection probability by the other ants and is called pheromone reinforcement in ACO algorithm.

If the pheromone value decreases slowly, ants will be trapped in suboptimal solutions in most of the cases. While if it decreases quickly, ants will not take advantage of gathered data by the other ants. Therefore, the pheromone evaporation rate has a direct impact on exploration and exploitation of paths in ant-based algorithms.

Exploration means finding a new path, while, exploitation means improving the current found path. Assigning an appropriate pheromone evaporation rate is needed to set the proper trade-off between these two factors. More discussion on this issue can be found in the study by Claes & Holvoet (2011). The pheromone update rule which includes both pheromone evaporation and reinforcement phases is given as Equation 2.2.

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}\ , \tag{2.2}$$

where $\rho \in (0, 1]$ is a constant value, named pheromone evaporation, and m is the number of ants. The amount of pheromone laid on link i and j by ant k is calculated using Equation 2.3.

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{f_k} & \text{if the } k^{th} \text{ ant traversed link (i, j),} \\ 0 & otherwise \end{cases} \tag{2.3}$$

where Q is a constant value and f_k is the cost of found route by ant k.

4) *Stopping procedure:* The ACO algorithm is completed by reaching a predefined number of iterations, and an ant is dropped by arriving at a predefined maximum number of hops before reaching its destination.

2.2.2.2. Types of ACO and its applications

AS is the first example of ACO algorithms which is proposed by Marco Dorigo, Maniezzo, and Colorni (1991) to solve popular travelling salesman problem (Flood, 1956). Promising results in its initial experiments and also its novelty attracted researchers' attention and a number of AS extensions were introduced in recent years that could improve the ASs performance in a significant way. These extensions include elitist AS (Marco Dorigo, 1992), rank-based AS (Bullnheimer, Hartl, & Strauss, 1997), and MAXMIN AS (Stützle & Hoos, 2000). Although all of these extensions use the same pheromone evaporation and solution construction mechanisms as original AS, the main differences between the original AS and these extensions are the pheromone updating procedure and some additional details in the pheromone trails management. There are some other ACO algorithms that more significantly modify the features of the AS in the literature. These algorithms including but not limited to Ant-Q (Gambardella & Dorigo, 1995), ant colony system (Marco Dorigo & Gambardella, 1997), ANTS (Maniezzo, 1999) and Hyper-cube AS (Blum & Dorigo, 2004). More details about these algorithms and their differences can be found in the study by Marco Dorigo and Stützle (2010).

In addition to proposing different extensions of the ACO algorithm, ACO has been widely utilized in different research areas and industries in recent years. Computer, electronic, civil and mechanical engineering are the most predominant domains that receive more benefits from ACO. In most of these areas, ACO has very near performance to the best algorithm performance or event has better performance compare with the existing best solution. Hence, this section gives a comprehensive overview of the most noteworthy applications of ACO. These applications are classified into six classes according to the problem types. These six classes are illustrated in Figure 2.4 and explained as follows:

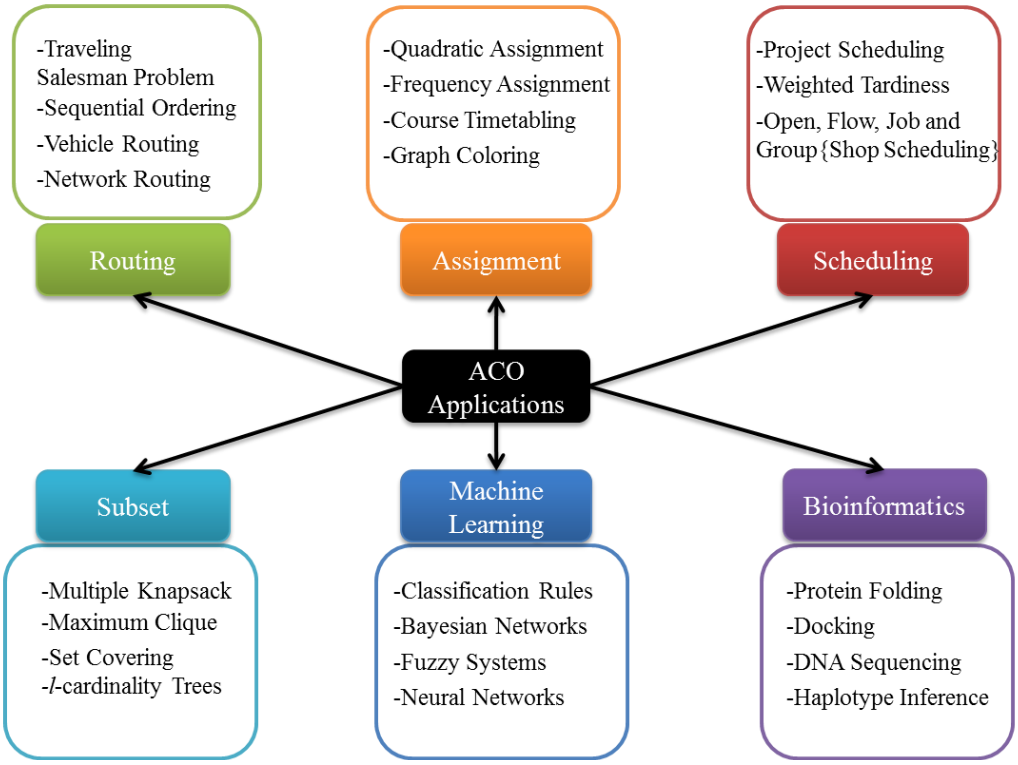

Figure 2.4. ACO Applications

1) *Routing:* Routing is one of the most important problems which is common among various domains. ACO uses one or more agents (ants) to explore the problem space, which is a graph in most of the cases, in order to find a path, between two points (e.g. source and destination) by considering different constraints such as time, distance or number of hops. Travelling salesman problem (Flood, 1956), sequential ordering (Gambardella & Dorigo, 2000), vehicle routing (Fuellerer, Doerner, Hartl, & Iori, 2009) and network routing (Di Caro & Dorigo, 2011) are the most notable examples for this problem domain.

2) *Assignment:* This problem domain is related to assigning members of a set (items) to members of another set (resources) considering certain rules and limitations. These sets can be objectives, agents, tasks or available resources (e.g. bandwidth, memory). ACO solves this problem using two decision stages. At first stage, the order of items (first set) is determined, while, resource (second set) assigning to items happens in second stage. Quadratic assignment (Stützle & Hoos, 2000), course time table (Socha, Sampels, & Manfrin, 2003), frequency assignment (Maniezzo & Carbonaro,

2000) and graph colouring (Costa & Hertz, 1997) are four well-known examples of this class.

3) Scheduling: Scheduling is related to allocation of scarce resources to tasks over time considering low cost of service providing. Scheduling problems have fundamental role in ACO research are and several type of scheduling problems have been solved by ACO algorithms such as project scheduling (Merkle, Middendorf, & Schmeck, 2002), weighted tardiness (Merkle & Middendorf, 2003), open shop, flow shop, job shop and group shop scheduling problems (Brucker & Brucker, 2007).
4) *Subset:* Solution for these kinds of problems consists a subset of available items by considering specific constraints for desired problem. Based on existing literature, ACO rarely used to solve subset problem. However, multiple knapsack (Leguizamon & Michalewicz, 1999), maximum clique (Solnon & Fenet, 2006), set covering (Lessing, Dumitrescu, & Stützle, 2004), L-cardinality trees (Blum & Blesa, 2005) are successful instances of ACO application in solving subset problems.
5) *Machine Learning:* Most of the existing problems in the machine learning area can be counted as optimization problems and both classic and meta-heuristic algorithms (i.e. ACO) can be used for solving these problems. Wide ranges of problems are exist in this problem domain, but, ACO algorithms have been applied to a few of these problems, namely classification rules (Otero, Freitas, & Johnson, 2008), Bayesian networks (De Campos, Fernandez-Luna, Gámez, & Puerta, 2002), fuzzy systems (Casillas, Cordón, & Herrera, 2000) and neural networks (Socha & Blum, 2007).
6) *Bioinformatics:* Similar to the machine learning research field, many problems in the bioinformatics field can be considered as optimization problems and solved by ACO algorithms. Although ACO rarely utilized for solving bioinformatics problems, protein folding (Shmygelska & Hoos, 2005), docking (Korb, Stützle, & Exner, 2007), DNA sequencing (Blum, Vallès, & Blesa, 2008) and haplotype inference

(Benedettini, Roli, & Di Gaspero, 2008) are successful instances of ACO application in solving bioinformatics problems.

2.3. Conclusion

This chapter discussed two main concepts, namely VANET and ant-based algorithm, which are the cornerstone of our proposed approach in this thesis. In addition to the concept of VANET, its applications and requirements and challenges are briefly reviewed in order to give a general overview to the readers. In order to provide an overview of ant-based algorithms, bio-inspired algorithms and their well-known examples (e.g. bee, bat and firefly) are briefly explained, since ant-based algorithms are a subset of these algorithms. The main steps and procedure of ACO along with various types of ant-based algorithms and their applications are summarized to provide comprehensive details regarding this type of algorithms. After providing a brief overview of VANETs and ant-based algorithms concepts, the existing VTRS are overviewed and classified in the next chapter.

CHAPTER 3: STATE-OF-THE-ART ON VEHICLE TRAFFIC ROUTING SYSTEMS

The most important distinction of VTRSs is whether the system utilizes a static or a dynamic routing algorithm for vehicle routing. Although both algorithms (i.e. static and dynamic) are used to route vehicles from their origins to their destinations, static algorithms do not consider the real-time traffic information of roads in their routing procedure. The algorithms that perform on graphs with fixed and pre-defined edge weights are called static routing algorithms, while, edge weights vary based on the real-time traffic information in dynamic algorithms. Hence, this chapter discusses about VTRSs by dividing these system into two classes based on their routing algorithms, namely static and dynamic VTRSs. After providing a brief overview of static algorithms and VTRSs in section 3.1, dynamic VTRSs are discussed in Section 3.2 with more concentration on bio-inspired based approaches in Section 3.2.1. Since ant-based algorithm is the cornerstone of our proposed approach, a comprehensive and statistical overview of ant-based VTRSs (i.e. a VTRS that uses ant-based algorithm for vehicle routing) is represented in Section 3.2.2. In addition, a taxonomy for ant-based VTRSs, which includes three main classes, namely, ACPs, ACPre, and ACS, is proposed in this section. Besides, statistical overview of studied ant-based VTRSs and major characteristics of VTRSs which should be considered in designing VTRSs are discussed in this section. Finally, section 3.3 concludes this chapter.

3.1. Static Routing Algorithm-based VTRSs

Over the years, several types of static routing algorithms have been proposed and used for finding shortest path between two or more points. The imposed requirements on the graph, execution time and roads' priority assignment policy are the main differences between various static routing algorithms. Various static routing algorithms are evaluated on real road maps by Zhan (1997), and Zhan and Noon (1998) in order to investigate their suitability to be used in VTRSs. Most of the existing vehicle routing systems (e.g. TomTom, Garmin and PAPAGO) use static routing algorithms to route the vehicles through the shortest distance path (B.-J. Chang, Huang, & Liang, 2008). Hence, an overview of the most well-known static routing including Bellman-Ford, A* and Dijkstra algorithms is provided in the following paragraphs.

Dijkstra algorithm (Dijkstra, 1959) is a popular static routing algorithm which is able to find the shortest path from a single-source node to the other nodes in a directed graph with non-negative edge weights. This algorithm uses greedy mechanism in a way that the local optimum (the edge with the lowest weight) is selected in each step until finding the shortest path from the source to destination (i.e. the global optimum). In this way, the search method of Dijkstra can be visualized as an expanding circle around the origin node that continues expanding in all directions until reaching to the destination. The running time of Dijkstra algorithm is $O(V^2)$ where V is the number of nodes in the graph.

The A* algorithm proposed by Hart, Nilsson, and Raphael (1968) to find the shortest path between two desired points in a graph without any negative edge weights. Heuristic estimate function is used for shortest path finding via A* algorithm in order to concentrate the searching procedure in the direction of the destination node. This concentration is performed by adding the heuristic estimate function value for the current and destination nodes to the final edge weigh via following function: $F[v] = h[v] + d[v]$ in which $d[v]$ and $h[v]$ indicate the current node's weight and heuristic estimation, respectively. $F[v]$ indicates the estimate of the best path via node v. The running time of A* algorithm is associated to the heuristic estimate function.

The Bellman-Ford algorithm is proposed by Ford and Fulkerson (1962) in order to find the shortest path between two points (i.e. source and destination) of a graph. One of the main features of this algorithm is that, unlike Dijkstra and A* algorithms, it can find the shortest path in a graph which has some edges with negative weight. The shortest path finding mechanism of this algorithm is similar to Dijkstra algorithm but operates an additional step to ensure that no negative path cycles are formed that prevent proper path finding. The running time of this algorithm is $O(V.E)$ where V and E indicate the number of nodes and edges, respectively, in the graph.

Although static routing algorithms are used in most of the existing VTRSs to propose the shortest paths to the drivers, these algorithms cannot provide an optimal routes for drivers due to lack of considering real-time traffic information. Dynamic VTRSs that uses real-time traffic information in their routing algorithm are proposed to overcome static VTRSs' problems. Hence, an overview of dynamic routing algorithm-based VTRSs is provided in the following section.

3.2. Dynamic Routing Algorithm-based VTRSs

It is worth noting that most of the recent approaches and solutions in vehicle traffic domain utilize the collected real-time traffic data via GPS, Geographical Information Systems, V2V and V2I communications to predict the future traffic topology and information on the roads (Gong, Li, & Peng, 2008). Predicted data can be used for appropriate vehicle routing and congestion avoidance. For instance, in the study by (Fu, 2001), the travel time of each road is modeled as a random variable and its realization can be predicted before the vehicle enter the road. An approximate probabilistic treatment is applied to the recurrent relations and a labeling mechanism is developed for solving the recurrent equations. A simple traffic flow model is developed and compared based on three different route choices, namely shortest distance, shortest time and shortest time with route information sharing by Yamashita, Izumi, and Kurumatani (2004). In the route information sharing system, each vehicle transmits route information (current position, destination, and route to the destination) to a route information server, which estimates future traffic congestion using this information and feeds its estimate back to each vehicle. Each vehicle uses the estimation to re-plan their route. Based on their experimental results, the average travel time of drivers using the route information sharing system is shorter than the time of drivers who chose shortest distance or simple shortest time estimates. Current navigation systems/applications that use dynamic algorithms for path finding (e.g. WAZE and INRIX) route the drivers based on minimizing each individual drivers' travel time without considering the impact of that routing on the travel time of other drivers. Although WAZE (WAZE, 2015) is a prominent social navigation application used by Malaysian in recent year, user preferences are neglected in the path selection procedure of this applications (Sha, Kwak, Nath, & Iftode, 2013).

A dynamic routing algorithm is presented by Yang et al. (2006) for enhancing the speed of calculating the optimum path. This algorithm is a kind of driver characteristic-based optimal path algorithm which is double hierarchical. An integrated model for managing dynamic route guidance and ramp metering via model predictive control is proposed by Karimi, Hegyi, De Schutter, Hellendoorn, and Middelham (2004). Dynamic route guidance and ramp metering are used as information provider and control tool, respectively. Drivers are able to choose alternative routes based on the provided information and delay is distributed over the highways via ramp metering control. Although the obtained results show that vehicles travel time is reduced by using this integrated model, the controller variables' optimal value and also comparing with other existing approaches are not considered and discussed. Neural networks are also utilized by some researchers (Dia, 2001; Hodge, Krishnan, Jackson, Austin, & Polak, 2011; Park et al., 2011; Shen, 2008; Van Lint, Hoogendoorn, & van Zuylen, 2005; W. Zhang, Zhang, & Xu, 2006) for vehicle travel time and speed prediction due to their learning capabilities. In these approaches, proposed algorithms trained with historical traffic data and used real-time traffic data for vehicle travel time and speed prediction. Vehicle travel time prediction is used in most of the existing approaches.

Traffic congestion recognition and avoidance approach by using vehicular networks is proposed by Wedel, Schunemann, and Radusch (2009). Average travel speed of vehicles is used for congested road detection and congestion avoidance. Three re-routing strategies, namely multi-path load balancing consider future vehicle positions, random multi-path load balancing, and dynamic shortest path, are presented by Pan, Khan, Sandu Popa, Zeitouni, and Borcea (2012), for vehicle congestion avoidance that use V2I communications for real-time data collection. Another travel time and traffic congestion prediction method using ad hoc networks is represented by Batool and Khan (2005). They developed a multilayer feed forward neural network combined with a back-propagation algorithm (Rumelhart, Hinton, & Williams, 1988) for traffic congestion prediction.

Congestion avoidance and route allocation using virtual agent negotiation (CARAVAN) (Desai, Loke, Desai, & Singh, 2013) is a multi-agent system that is developed for finding optimal paths within a short time and with low communication overheads. Vehicles exchange preference information and use virtual negotiation for collaborative route allocation through inter-vehicular communications in CARAVAN. Another multi-agent system is proposed by Chen, Yang, and Wang (2006) in order to enhance urban traffic control system. For this aim, an agent-based distributed hierarchy traffic control system is proposed and integrated with VTRS by the authors. A multi-agent based VTRS architecture is developed by Shi, Xu, Xu, and Song (2005) in which sensor networks are used for collecting real-time traffic information. In addition, integration of radio frequency technology and optical fiber network is proposed in order to mitigate the sensor networks' communication drawbacks.

Another point that should be mentioned is that dynamic routing algorithms can be divided into two classes, namely infrastructure-based and infrastructure-free. In the former case, a number of cameras, loop detectors, sensors and communication devices should be accommodated throughout the roads for monitoring and real-time data collection purposes by taking advantage of V2I and V2R communications. The gathered data is transferred to a traffic information center for further processing. An on-demand request can be sent by each vehicle to traffic information center for achieving desired information. The proposed approaches by Shi et al. (2005), Yamashita et al. (2004) and Xuedan Zhang et al. (2007), can be considered as examples of infrastructure-based dynamic routing algorithms. Another infrastructure-based traffic information system, called PeerTIS, developed by Rybicki, Scheuermann, Koegel, and Mauve (2009) in which IP-based communication channel is used by vehicles to establish a peer-to-peer overlay over the Internet. Existing cellular Internet access is the cornerstone of PeerTIS. This system is enhanced in another study by Rybicki, Pesch, Mauve, and Scheuermann (2011) in which graph is used for road map representation and this graph is divided into sub-graphs when a new vehicle enters the system. In addition, publish/subscribe mechanism (Eugster, Felber, Guerraoui, & Kermarrec, 2003) is used to handle the updated information efficiently. The mentioned infrastructures and equipment for infrastructure-based dynamic routing algorithms are not required in infrastructure-free dynamic routing algorithms. These approaches take advantage of V2V communications and multi-hop relay networks for real-time data gathering. Some examples of this approach are as follows: Self-Organizing Traffic Information System is developed by Wischoff, Ebner, Rohling, Lott, and Halfmann (2003) in which every vehicle monitors its surrounding area to gather local traffic information, and also receives data packets including detailed information from neighboring vehicles. In this way, each vehicle processes its obtained data in order to find its desired path and also broadcast the result to all neighboring vehicles. Another V2V-based traveler information system along with a location-based broadcasting protocol are designed by Zhang, Ziliaskopoulos, Wen, and Berry (2005). In addition, another infrastructure-free approach is proposed by Jerbi, Senouci, Rasheed, and Ghamri-Doudane (2007) for road density prediction. In this approach, every road is divided into a number of segments and segment density data packet used for gathering vehicle density on each road. This information can be used by vehicles in order to avoid congested roads in their path selection procedure. Vehicle-to-Vehicle Real-time Routing (Ding, Wang, Meng, & Wu, 2010) is another example of second class. It includes two algorithms: 1) guidance algorithm which discovers all existing path between source and destination by utilizing an intelligent and limited-scope flooding mechanism. 2) detour algorithm which utilizes RDP and LDP routing to lay away the areas comprising empty roads. Another infrastructure-free VTRS is developed by He, Cao, and Li (2012) in which density-speed flow model is used for traffic situation prediction. This approach uses a dynamic candidate path selection mechanism in order to decrease the redundant data gathering overhead. Besides, an adaptive vehicle routing system based on wireless sensor networks is developed by Chang et al. (2008). This approach takes advantage of WiMAX multi-hop relay networks to enhance the reliability and efficiency of V2V communications. Similar to all infrastructure-free approaches, vehicles themselves are used for data gathering. Vehicle density, road class and distance can be obtained via gathered data and be used in path finding procedure. Although prediction of traffic information such as vehicle density, travel speed and time are important factors for vehicle traffic routing, it is a difficult process due to various unpredictable events and dynamic nature of vehicular environments. More information regarding VTRSs and their classifications can be obtained in the study by Khanjary and Hashemi (2012).

Although most of the mentioned approaches obtained promising results for reducing travel time, improving traffic flow and efficient vehicle routing, various criteria such as distance, traffic load, road width, risk of collision, number of intersections, weather condition, dynamic changes of vehicular environments and special events in path selection procedure are not considered simultaneously in their routing procedure. As mentioned in Chapter 1, this problem is known as multi-criteria shortest path problem and bio-inspired-based VTRSs have been widely used to overcome this problem and the obtained results are promising (Teodorović, 2008; Tonguz, 2011). Hence, an overview of bio-inspired-based VTRSs is provided in the following section.

3.2.1. Bio-inspired-based VTRSs

Although most of the existing bio-inspired algorithms are used to solve vehicle traffic routing and congestion problems, ant, bee, genetic and PSO are most commonly used algorithms among researchers over the years. Hence, this section gives an overview of these four algorithms. However, since the ant-based algorithm is the cornerstone of our proposed framework, ant-based VTRSs are discussed in more details in Section 3.2.2.

a) Bee-based VTRSs: Several bee-based algorithms, such as virtual bee (Yang, 2005), BeeAdHoc (Wedde & Farooq, 2005), marriage in honeybees (Abbass, 2001), BeeHive (Wedde, Farooq, & Zhang, 2004), bee colony optimization (Teodorović & Dell'Orco, 2005) and artificial bee colony (Karaboga, 2005), have been developed by researchers during the last decade. Originally, bee algorithm was proposed for solving numerical problems. Thus, the first studies by Karaboga and Akay (2009), Karaboga and Basturk (2007, 2008), and Krishnanand, Nayak, Panigrahi, and Rout (2009) aimed to evaluate its performance on numerical benchmark functions and to compare it with other well-known bio-inspired algorithms such as ant, genetic and PSO. Bee-based algorithms are widely utilized in various areas of research in the recent years. Not only Bee-based algorithm could be used in theoretical aspect but also its role in real world problems and industry is undeniable. Mechanical, electrical, civil and software engineering are some research areas that take advantages from bee-based algorithms. A comprehensive study of bee-based algorithms and their applications can be found in the study by Karaboga, Gorkemli, Ozturk, and Karaboga (2014). In the case of bee-based algorithms application in VTRSs, a bee-based methodology is developed by Teodorović and Dell'Orco (2008) to solve the ride-matching problem and consequently vehicle traffic congestion problem. Ride-matching is one of the most popular examples of travel demand management (Teodorovic, Edara, & Via, 2005) strategies. It helps two or more commuters to share a vehicle while traveling from their origins to destinations in order to reduce the number of vehicles with only one passenger and also to mitigate vehicle congestion. Saving money, reducing stress and travel time are some other advantages of this approach. Wedde and his colleagues started to work on a new decentralized and self-organized bee-based VTRS over the years. They have published a series of conference and journal papers about their system. The initial idea is represented by Wedde et al. (2007) and called Bee-inspired Jam Avoidance (BeeJamA) system. The cornerstone of BeeJamA is BeeHive routing algorithm which was proposed for large computer networks by Wedde et al. (2004). BeeJamA is a multi-agent VTRS that is proposed to minimize vehicles' travel time by preventing traffic jam conditions. BeeJamA gets up-to-date and instant traffic information using V2I communications like floating car data (Pfoser, 2008). The cornerstone of these communications is a decentralized network, namely navigators. Navigators are responsible for a specific area which is called navigation area, where it communicates with its own area's vehicles and sends routing instructions to them. A hop by hop routing takes place for each vehicle via received guidance data for next intersection before reaching current intersection. This procedure is explained in more details in the study by Senge and Wedde (2010), and Wedde and Senge (2013). To achieve real-time hop by hop vehicle routing guidance for large-scale vehicle traffic scenarios, a 5-layered bottom-up model is proposed by Wedde and Senge (2013). Physical, routing, area, net and communication are these five layers. BeeJamA's performance is compared against A*-based algorithms relaying on global information systems (e.g. Dynamic Shortest Path) by Senge and Wedde (2012a). The obtained results show that it outperforms all A* algorithms under all degrees of penetration rates as well as considering reactive flexibility and easy scalability. In addition, BeeJamA decreased average traveling time of vehicles and also there is inverse relationship between the accounted vehicles' traveling times and BeeJamA penetration rate. BeeJamA is introduced as an example of Cyber-Physical System by Senge and Wedde (2012b). Cyber-Physical System is defined as: "it is requested that the structures and relationships of the software and physical/real world layers be as similar as possible, ideally even isomorphic or congruent" (C. H. Liu et al., 2014; Senge & Wedde, 2012c). Concurrency, real-time capability, decentralized control, self-adaptation, self-organization, reliability and fault tolerance are the most common requirements and ends of Cyber-Physical Systems (Senge & Wedde, 2012b). Although BeeJamA could mitigate the recurring congestion considerably, non-recurring congestion conditions cannot be handled in a significant way. Another bee-based vehicle traffic optimization and management approach is proposed by Ghosal, Chakraborty, and Banerjee (2013). Maximum speed utilization of vehicles, planning lanes for unplanned traffic, each lane's speed and their differences are considered in this bee-based algorithm in order to mitigate vehicle traffic congestion and also handling vehicle routing. From another point of view, the way that this system is utilized can be a potential drawback of this system. This is because it increases the number of lane transitions to reach the optimum high speed which itself can lead to traffic congestion. Recently, a bee-based zonal VTRS is developed by L. Wu, Yang, Liu, and Zhang (2014) in which the road map is divided into various zones based on

Shapley value game (Narayanam & Narahari, 2010). Then, V2V and V2I communications are used for real-time traffic data gathering in each zone. Finally, bee-based algorithm is utilized to route the vehicles through the optimal path from origin to destination inside each zone. The main drawback of this approach is that it needs to adjust several parameters such as Euclidean distance, minimum similarity and road section number thresholds.

b) PSO-based VTRSs: PSO is another bio-inspired algorithm which is used for both traffic flow prediction and vehicle routing. PSO algorithm is utilized by Mohemmed, Sahoo, and Geok (2008) to solve shortest path problem. A priority-based indirect path encoding strategy and a heuristic operator are used for widening the search area and preventing loop creation in path finding procedure, respectively. After that, a PSO-based VTRS is proposed by Qun (2009). In addition to analyze the characteristics of both VTRSs and PSO algorithm, PSO application in path finding is explained and examined for the first time. PSO efficiency and performance in path finding and searching is evaluated through a simulation example. However, PSO can easily get into local optimum which is not considered in the mentioned approach. Hence, Deng, Tong, and Zhang (2010) proposed a hybrid approach (i.e. combination of PSO and fluid neural network) in order to overcome the PSO drawback of falling into local optimum while finding shortest path in vehicular environments. In addition, the weight coefficient symmetry restrictions of the traditional fluid neural network is solved by this hybrid PSO-based VTRS. Examining the proposed approach in a small road network (i.e. 20 nodes) is its main drawback (Kammoun, Kallel, Casillas, Abraham, & Alimi, 2014). Regarding traffic flow prediction, analysis of common traffic flow predictive model is used by Peng (2011) to propose a PSO-based combined traffic flow prediction. In order to overcome the impractical time-invariant assumptions which are the cornerstone of the most existing traffic flow predictors, an intelligent PSO is developed by Chan, Dillon, and Chang (2013). The proposed intelligent PSO is a hybrid approach in which PSO, neural network and fuzzy system are integrated in order to adapt the time-variant characteristics of traffic flow and also configurations of on-road sensors systems. The effectiveness of this approach is examined and proved by comparing its predicted traffic flow with real traffic flow obtained by the on-road sensor system. Recently, another hybrid traffic flow prediction approach is introduced by Hu, Gao, Yao, and Xie (2014) in which PSO and support vector regression are integrated. Prediction model is established via support vector regression and the model's parameters are optimized by utilizing PSO algorithm. The obtained results show that this PSO-based approach outperforms multiple linear regression and Back-Propagation neural network (Ng, Cheung, Leung, & Luk, 2003) in term of traffic flow prediction. It is worth noting that a comprehensive survey on PSO and its applications can be found in the study by Lalwani, Singhal, Kumar, and Gupta, (2013).

c) Genetic-based VTRSs: Using genetic-based VTRSs for solving route planning have been reported over the years (Yu & Lu, 2012). GA is used for the first time by Gen, Cheng, and Wang (1997) to solve the shortest path problem. They proposed a priority-based encoding strategy for finding all possible paths in a graph. Although the same chromosome length is selected and the encoding process is so complicated in this approach, it was a good starting point. Afterwards, some researchers (Jun Inagaki, Miki Haseyama, & Hideo Kitajima, 1999; Kanoh & Nakamura, 2000; Leung, Li, & Xu, 1998) used GA to find multiple routes in VTRSs. However, they did not pay attention to paths overlap in multiple path finding. Although Inagaki, Haseyama, and Kitajima (1999) designed a genetic-based algorithm to minimize the number of overlapped paths in multiple path finding mechanisms, large solution area is needed for obtaining high quality solution because of inconsistent crossover strategy utilization. One of the well-known genetic-based algorithms for solving shortest path problem is proposed by Ahn and Ramakrishna (2002). Variable-length chromosome and a repair strategy are used for path representation and preventing loop creation in path finding procedure, respectively. Based on the simulation results, their proposed approach outperforms the other genetic-based approaches in terms of route failure ratio and convergence rate for small and moderate size networks. A genetic-based algorithm with a part of an arterial road regarded as a virus is proposed by Delavar, Samadzadegan, and Pahlavani (2004) to find an optimal path between origin and destination under static environment. Although search rate is enhanced via crossover and virus evolution theory, mutation was not utilize in this approach. Another genetic-based VTRS for static environments is represented by Alhalabi, Al-Qatawneh, and Samawi (2008). They proposed a novel selection method based on choosing the best next neighbor node and compared it with tournament selection method. The effects of various mutation points and number of crossover points on VTRSs are also examined in this study. Since, the vehicular environments are highly dynamic, Davies and Lingras (2003) proposed a genetic-based algorithm for dynamic environments. They changed the shortest path problem into the shortest walk problem. Crossover is used when the walking condition shifted to bad (i.e. walk toward the previously seen node). All the discussed approaches concentrate on single objective problems and evaluated only with small size networks. It is worth noting that Nanayakkara et al. (2007) developed a genetic-based route planning algorithm which can handle the routing procedure on large scale networks (i.e. maps with thousands of nodes).

In the case of multi-objective shortest path problem, a new fitness function for finding multi-objective optimal paths with minimum overlap for VTRSs is proposed by Chakraborty and his colleagues (Chakraborty, 2004, 2005; Chakraborty, Maeda,

& Chakraborty, 2005). Some criteria such as distance, number of turns, passing through the mountain are used for alternative path selection and also penalty for fitness. Some other multi-objective shortest path approaches based on GA presented by Kanoh (2007), Kanoh and Hara (2008) in which various objectives (e.g. distance, travel time and number of traffic light) are considered in path finding procedure and the initial population is obtained via Dijkstra algorithm. A multilayer hierarchy network strategy (Wen & Gen, 2008) is developed to significantly decrease the required computation time for path finding in large road maps. A genetic-based cluster method is embedded in this strategy to mitigate the size limitations with high accuracy rate. A hybrid multi-objective VTRS which uses GA and λ–interchange local search method combination and evolved from improved A* algorithm is proposed by Hu, Gu, Huang, Yang, and Song (2008). Another hybrid approach is designed by Chakraborty and Chen (2009) in which GA and fuzzy system are integrated to find optimal alternative paths based on drivers' requirements. In this approach, the found paths are modified via feedback mechanism. A multi-objective mathematical formulation for vehicle routing along with four main objectives (i.e. distance, safety, total time and cost) is discussed in the study by Kim, Jo, Kim, and Gen (2009). In this genetic-based approach, hash, adaptive-weight and priority-based encoding methods are utilized for selection method, finding solution sets and representing chromosome, respectively.

Another genetic-based multi-objective VTRS which uses driving distance, time and cost as objectives in route finding procedure is developed by Wen, Gen, and Yu (2011), and Wen and Lin (2010) and called distance Pareto GA. Two-level road map hierarchy is used in order to decease the computational time. A new fitness function based on two types of distance values, namely Pareto distance and crowding distance, is introduced and used in this approach. DGA (Lee & Yang, 2012) is a hybrid re-routing algorithm which, similar to the approach proposed by Kanoh and Hara (2008), integrates Dijkstra and GA. The main difference between these two approaches is the usage of GA in these approaches. In the former, GA is used for considering various criteria in route finding process, while, DGA algorithm utilizes GA for re-routing purposes. Yu and Lu (2012) proposed one of the most notable genetic-based VTRSs in which crossover and mutation operators are redefined in single mode and two novel operators, namely hyper-crossover and hyper-mutation, are designed as inter-mode operation. A new fitness function based on p-dimensional vector (i.e. for considering multiple objectives) is used for optimal path finding. Similar to other genetic-based approaches, routes are represented by variable length chromosomes, while, sub-chromosomes are used to define various types of transportation modes.

An improved version of GA is proposed and used for finding optimal path by En et al. (2012). The population set is optimized by omitting the unnecessary nodes and paths before algorithm initialization. In each generation, the population's best individual is protected via integration of roulette and elite protecting methods. This approach outperforms the others in terms of convergence rate and solution quality. A novel genetic-based VTRS which uses Petri net analysis as fitness function is developed by Dezani et al. (2014). Petri net analysis enables the system to control the whole road network in a real time. VANET infrastructures are used for gathering real-time traffic data in this approach. Most recently, a genetic-based methodology is proposed by Cagara, Bazzan, and Scheuermann (2014) to enhance the optimal usage of existing roads by distributing vehicles through alternative paths. Although this methodology is evaluated in terms of network performance and convergence speed, it is not compared with other existing approaches. A comprehensive and statistical overview of ant-based VTRSs is represented in the following section.

3.2.2. Ant-based VTRSs

A comparative study and taxonomy of ant-based VTRSs are provided in this section. In our proposed taxonomy, ant-based VTRSs are classified into three main classes based on the method or strategy that is used in these approaches to overcome vehicle routing and congestion problems. ACPs, ACPre, and ACS are these three classes. Many parameters in ant-based approaches depend on problem characteristics and searching strategies to find the problem space. Therefore, most studies are conducted in ACP to achieve the best value set for ant-based VTRSs. Considering the dynamicity of topology and traffic in vehicular networks, the prediction of upcoming network conditions is a necessary task. Thus, researchers have also focused on ACPre. In addition, some researchers proposed map segmentation to reduce the computation time of the algorithm, which are discussed in ACS class of our proposed taxonomy. Each of these classes along with their relevant approaches is discussed in the following subsections.

3.2.2.1. Ant Colony Parameters (ACP)

This section discusses ant-based VTRSs that exploit the basic concept of ant colony algorithms. Researchers have changed the pre-existing variables and steps of the original ant algorithm (Marco Dorigo, Maniezzo, & Colorni, 1996) without adding any new concepts. The two types of ACP (i.e., variables and steps) are discussed as follows.

a) *Variable-based ACP:* Several key variables such as number of ants (n), pheromone power (α), heuristic power (β), pheromone evaporation (ρ), and ant speed (v) affect the ant-based approaches. If these variables are not properly assigned, the ants will follow a previously found path that is not necessarily optimal because of the dynamic topology of vehicular networks. The assignment of appropriate values to these variables allows ants to search the route map to find the optimal path quickly and accurately. Liu et al. (J. Liu et al., 2007) proposed a variable-based algorithm that uses the JIN method (Jin, Hong, & Gao, 2002) on ant-based algorithms for path routing optimization. Based on their findings, the convergent speed of search procedures is directly related to the above mentioned variables and the probability of choosing a non-optimal path is equal to the convergence rate. Ok, Seo, Ahn, Kang, and Moon (2011) proposed another variable-based short path-selection algorithm based on map link properties. Their findings show that increasing the number of ants reduces the probability of discovering non-preferred paths. Thus, the overall path length will be increased and the algorithm will converge toward the same path. However, a small number of ants cannot cause the above mentioned convergence rate. A traffic congestion control method based on different preferences is developed by Nahar and Hashim (2011). These preferences allow the algorithm to reduce average travelling time by adjusting ant colony variables. Their results show that the number of ants is directly correlated with the algorithm performance. The number of ant agents should not be less than the threshold defined in the algorithm (Nahar & Hashim, 2011).

b) *Step-based ACP:* Unlike the variable-based ACP, wherein the main concern is the assigning of the best values to the variables, the step-based ACP concentrates on enhancing the steps of ant-based approaches including the following steps: ant distribution initialization, ant probability function, pheromone updating, and the stopping phase. These steps aim to enhance the algorithm performance. Most studies focused on the first two steps of ant-based algorithms because they have the most impact on these approaches as follows:

1) *Initialization step:* Multi agent system (MAS) has been reported as a promising approach for dynamic problems (wherein involved parameters are not constant and can be change dynamically) because MAS contains common features between the swarm behavior of agents (e.g., ants, bees, bat, birds, or fish) and vehicular ad hoc networks (García-Gonzalo & Fernández-Martínez, 2012; Kponyo, Kuang, & Li,

2012). MAS is composed of a number of independent agents that are located in the problem space in a decentralized manner to solve dynamic problems. Ant agents have proven to be superior to the other agents in many studies such as (Bonabeau et al., 1999; S. S. Dhillon & P. Van Mieghem, 2007; Di Caro & Dorigo, 2011; Schoonderwoerd, Holland, Bruten, & Rothkrantz, 1997). (Weyns, Holvoet, and Helleboogh (2007) proposed different types of cooperative ant agents, such as intention and exploration agents. They used a divide and conquer route map to reduce traffic congestion. Exploration agents investigate the environment, whereas intention agents are used to allocate road segments to different vehicles. Foroughi, Montazer, and Sabzevari (2008) proposed a modified ant-based algorithm that minimizes congestion time. They also used a path with minimum traffic and length to optimize travel time, fuel consumption, and air pollution. Kammoun, Kallel, Alimi, and Casillas (2010) proposed an adaptive vehicle guidance system that intelligently finds the best route by using real-time changes in the network. To achieve dynamic traffic control and improve driver request management, this method used three types of agents, namely, city agent, road supervisor agent and intelligent vehicle-ant agent. A multi-agent evacuation model was introduced by Zong, Xiong, Fang, and Li (2010) to minimise the total evacuation time for vehicles and balance traffic load. Experiments have shown that MAS is more effective than a single agent system. Cong, De Schutter, and Babuska (2011) developed a model to optimize dynamic traffic routing by using a two-step approach: network pruning and network flow optimization. In the network-pruning phase, ant pheromone is removed after the best route is found by the agents to increase exploration rate. In the flow optimisation phase, which is based on ant-based algorithm with the stench pheromone and collared pheromone, the agents correspond to the links selected in the network-pruning phase only. Moreover, this two-step approach reduces the computational burden by addressing complex, dynamic traffic control problems. Król and Mrożek (2011) proposed a tool to investigate the existing traffic flow of a specific city (i.e., Wroclaw) to optimize the vehicle congestion problem. An ant-based algorithm is used to simulate vehicle conditions in the route network. Sur, Sharma, and Shukla (2012) proposed a multi-breeded ant-based VTRS that defines different ant agent groups instead of on agent group. In this system, each ant agent only follows agents of the same type (breed). This type of classification has a significant contribution to the distribution of vehicles. A multi-breeded ant-based VTRS allows the pheromone update to be independent of time and incorporates several dependent factors in the update. Ghazy, EL-Licy, and Hefny (2012) proposed a new threshold-based ACO by using roads' good travel times as threshold values to decrease the computation time. They introduced a new ant, called check ant, which is used to preserve the best path and discard the degraded routes. Bura and Boryczka (Boryczka & Bura, 2013; Bura & Boryczka, 2010, 2012) proposed a parallel version of dynamic fuzzy logic-ant colony system-based route selection system (Salehinejad & Talebi, 2010) by using a new type of pheromone update that occurs locally and globally. This method also returns blocked ants to the source instead of eliminating them at the destination.

2) *Probability function*: Ants select the next hop in the problem graph (e.g. road map) by using the probability function. This function can be categorized into two classes, namely, probabilistic and heuristic models (Kammoun et al., 2010). Hallam, Hartley, Blanchfield, and Kendall (2004) introduced ant behavior-based search agents called soft cars and used these agents to implement and test the model on a road network. The soft cars choose paths with less loads, shorter distances, and more lanes, as well as paths that are frequently visited by soft cars. The dynamic system for avoiding traffic jams (Bedi, Mediratta, Dhand, Sharma, & Singhal, 2007) was designed by adopting an alternative path for each selected solution (route). This

alternative path is used whenever the selected route is congested. The probability function of an ant algorithm is extended by using random function. Ge, Wang, Wang, and Jiang (2011) developed a crossing traffic rule-based VTRS by combining traffic rules with the probability function. This method considers the restriction and delay of direction, satisfies the requirements of actual traffic environment, and enhances the validity of vehicle paths. The distributed intelligent traffic system Kponyo et al. (2012) uses vehicle average speed as a parameter to determine the traffic condition. This system guides cars to paths with low traffic; thus, this system selects the best path more efficiently in compare with scenario where the ants select their path randomly. Table 3.1 summarizes aforementioned ACP approaches along with their objectives.

Table 3.1. Ant-based VTRSs considering ACP

Ant-based VTRS category	**Title of Paper (Ref.)**	**Objectives**
Variable-based		
	Ant Colony System Algorithm for Path Routing of Urban Traffic Vehicles (Liu et al., 2007)	To find an optimal path considering time and distance.
	An Ant Colony Optimization Approach for the Preference-Based Shortest Path Search (Ok et al., 2011)	To propose a preference-based shortest path.
	Modeling and Analysis of an Efficient Traffic Network Using Ant Colony Optimization Algorithm (Nahar & Hashim, 2011)	To create an optimum traffic system and a platform for vehicle congestion control.
Step-based		
Initialization	Anticipatory Vehicle Routing using Delegate Multi-Agent Systems (Weyns et al., 2007)	To propose a multi-agent systems, for anticipatory vehicle routing to avoid traffic congestion.
	Designing of a new urban traffic control system using modified ant colony optimization approach (Foroughi et al., 2008)	To minimize the congestion time by global management over most trips done in the under control area.
	An Adaptive Vehicle Guidance System instigated from Ant Colony Behavior (Kammoun et al., 2010)	To utilize real-time traffic information to increase the global velocity on the road network.
	Multi-ant Colony System for Evacuation Routing Problem with Mixed Traffic Flow (Zong et al., 2010)	To tackle evacuation routing problem with mixed traffic flow.
	A New Ant Colony Routing Approach with a Trade-off Between System and User Optimum (Cong et al., 2011)	To find the optimal distribution of traffic flows in the road network.

Table 3.1. Ant-based VTRSs considering ACP (continued)

Ant-based VTRS category	Title of Paper (Ref.)	Objectives
Step-based		
Initialization	Swarm-based Multi-agent simulation: A case study of urban traffic flow in the city of Wroclaw (Krol & Mrozek, 2011)	To develop a model of road traffic environment which can be used to optimize traffic flow.
	Analysis & Modelling Multi-Breeded Mean-Minded Ant Colony Optimization of Agent Based Road Vehicle Routing Management (Sur et al., 2012)	To find the optimal distribution of traffic flows in the road network.
	Threshold based AntNet algorithm for dynamic traffic routing of road networks (Ghazy et al., 2012)	To propose threshold based AntNet algorithm for vehicle congestion problem.
	Ant Colony Optimization for Multi-criteria vehicle navigation problem (Boryczka & Bura, 2013)	To propose user-preference vehicle navigation system utilizing multi-agent ant-based algorithm.
Probability Function	Optimization In a Road Traffic System Using Collaborative Search (Hallam et al., 2004)	To find the best path in a crowded city.
	Avoiding traffic jam using Ant Colony Optimization A novel approach (Bedi et al., 2007)	To choose an alternative optimum path to avoid traffic jam and then resume that same path again when the traffic is regulated.
	Urban Vehicle Routing Research Based on Ant Colony Algorithm and Traffic Rule Restriction (Ge et al., 2011)	To develop a vehicle path planning based on ant colony algorithm by considering crossing traffic rule restriction.
	Real Time Status Collection and Dynamic Vehicular Traffic Control Using Ant Colony Optimization (Kponyo et al., 2012)	To find an alternative optimum path to avoid traffic jam using average speed of vehicles at different roads.

3.2.2.2. Ant Colony Segmentation (ACS)

A problem space is divided into several less complex problems. The main idea of ACO segmentation is derived from divide and conquers approaches. Tatomir, Kroon, and Rothkrantz (2004), Tatomir and Rothkrantz (2004, 2006) proposed a Hierarchical Routing System (HRS) based on the ant algorithm. The hierarchical ant-based control algorithm (Tatomir & Rothkrantz, 2005) is combined with a HRS to increase scalability. Narzt et al. (2010) introduced another technique that uses segmentation as a principle to overcome traffic control problems. In this approach, a novel pheromone update with a user-preference assignment system is adopted to divide the environment into different clusters. The segmentation procedure is conducted by using a pheromone engine, and a unique identifier is assigned to each car. Claes and Holvoet (2012) introduced a cooperative ant-based algorithm that results in less iteration. The essential concept for cooperation in this approach is the concept of the region. According to this concept, ants are interested in paths that lead to locations near their destination. To achieve a suitable form of segmentation, segments near each other are grouped together to form a region. Thereafter, routing is performed according to regions instead of segments. Claes and Holvoet evaluated cooperative and non-cooperative approaches and concluded that cooperative methods outperform non-cooperative methods in different aspects. For instance, the number of ants required in cooperative methods is less than the number of ants in non-cooperative methods. Table 3.2 summarizes aforementioned ACS approaches along with their objectives.

Table 3.2. Ant-based VTRSs considering ACS

Ant-based VTRS category	**Title of Paper (Ref.)**	**Objectives**
Segmentation	Hierarchical routing in traffic using swarm-intelligence (Tatomir & Rothkrantz, 2006)	To develop a prototype of HRS by splitting traffic networks into several smaller and less complex networks.
	Self-organizing congestion evasion strategies using ant-based pheromones (Narzt et al., 2010)	To investigate a technical implementation of swarm intelligence applied to the traffic system and evaluates different evasion strategies for vehicles.
	Cooperative Ant Colony Optimization in Traffic Route Calculations (Claes & Holvoet, 2012)	To propose a cooperative ACO for vehicle traffic route calculation.

3.2.2.3. Ant Colony Prediction (ACPre)

The two kinds of prediction methods are long-term prediction and the short-term prediction. Given that the control parameters of traffic congestion change briskly and that vehicles are highly dynamic, long-term prediction is imprecise for traffic problems. Therefore, most techniques in this area use short-term predictions. Ando et al. (Ando, Fukazawa, Masutani, Iwasaki, & Honiden, 2006; Ando, Masutani, et al., 2006) proposed a basic model for predicting traffic congestion by using the probe car system. The probe car system is a data collection method that uses vehicular ad hoc networks to collect real-time traffic information. Based on their investigation, the distance-based pheromone update mechanism outperforms other types of update mechanisms, such as braking and basic traffic pheromones. Tatomir, Rothkrantz, and Suson (2009) proposed ant-based VTRS that applies the ant algorithm to find the fastest path by using past, present, and future traffic information (i.e., travel-time prediction). To obtain accurate data in this routing system, the number of ant agents should be more than the number of cars. Claes and Holvoet (2011) also used link travel-time prediction to find paths with the shortest travel times. Kurihara (2013) and Kurihara et al. (2009) proposed a novel congestion-forecasting algorithm that is composed of major phases. First, the flow of traffic density is formulated by using a traffic density pheromone. Thereafter, based on the congestion-diffusion concept, the growth of the path queues is calculated, thus enabling the congestion forecasting of pheromones by monitoring the evaporation rate. Ant colony routing (Cong, De Schutter, & Babuška, 2013) is another prediction based VTRS which uses stench pheromone and color ants for vehicle routing and congestion mitigation. The stench pheromone is utilized for distributing vehicles through the road maps; meanwhile, multiple origins and destinations can be represented by different colors and color ants are only sensitive to their own color. Moreover, network pruning method is used to omit the unnecessary links and nodes in order to decrease the road network complexity as well as computation time of the algorithm.

a) Hybrid techniques: Some studies have combined ant algorithm with techniques such as fuzzy logic, neuro-network, and machine learning to create hybrid techniques. The most applied techniques are as follows: Fuzzy logic can be embedded to form a multi-preference routing system and can be applied to pheromone update procedures to detect the optimum multi-objective direction (e.g., number of traffic lights, lane width, and accident risk) between sources and destinations. Neural networks are used to predict the future time by using real-time traffic information. The main advantage of using machine learning in ant-based algorithm is the ability of using passive information as learning input to optimize or predict future traffic conditions more effectively. Salehi-nezhad and Farrahi-Moghaddam (2007) and Salehinejad and Talebi (2008, 2010) combined fuzzy logic, neural network and ant-based algorithm to introduce a user-preference VTRS. In these methods, the traffic control center and artificial neural network obtain current and future traffic data, respectively. In addition, fuzzy logic is used for local pheromone updating. Abbas, Khan, Ahmed, Abdullah, and Farooq (2011) proposed another neuro-fuzzy and ant algorithm amalgamation to find the most encouraging route based on driver preference. Similar to the previous methods, artificial neural network is supplied to predict time. This system is able to prevent recurring congestion conditions and generate a priority-based path list for drivers. An adaptive VTRS that integrates ant algorithm with hierarchical fuzzy model is proposed by Kammoun, Kallel, Alimi, and Casillas (2011) and Kammoun et al. (2014). Their proposed approach includes two steps: 1) ant algorithm is used to find the best path taking into account both traffic quality and itinerary length, 2) hierarchical fuzzy model is used for enhancing the path selection considering a set of the most important factors regarding the driver, the environment and the path. Jiang, Wang, and Zhao (2007) proposed an ant-based VTRS that determines an optimal path (called the 'closest path' in the Dijkstra algorithm) by modifying pheromone updates and learning strategies. Yousefi and Zamani (2013) developed a learning-based VTRS based on ant algorithm concept. They considered that a network graph was composed of several sub-graphs. Learning the overall condition of the graph is possible by conducting several searches over each sub-graph. This procedure helps discover the shortest path and avoid choosing paths randomly. Time-ANTS (Doolan & Muntean, 2014) is another prediction-based VTRSs that uses machine learning to identify and avoid bottlenecks. Time-ANTS architecture includes four components, namely vehicle models, road models, a time-dependent traffic model (i.e. historical traffic data) and a current traffic model (i.e. real-time traffic data), for optimal path finding. Similar to most of the existing approaches, ignoring non-recurring congestion conditions and testing the algorithm in an unreal and small size road map network are the most notable drawbacks of this approach. Table 3.3 summarizes aforementioned ACPre approaches along with their objectives.

Table 3.3. Ant-based VTRSs considering ACPre

Ant-based VTRS category	**Title of Paper (Ref.)**	**Objectives**
Prediction		
	Pheromone Model Application to Traffic Congestion Prediction (Ando, Masutani, et al., 2006)	To propose a short-term prediction model for vehicle congestion control.
	Travel time prediction for dynamic routing using ant-based control (Bogdan Tatomir et al., 2009)	To propose a dynamic routing system based on Ant-Based Control using travel time prediction.
	Ant Colony Optimization applied to Route Planning using Link Travel Time predictions (Claes & Holvoet, 2011)	To find routes that reduce vehicles travel time using link (road) travel time predictions.
	Traffic-Congestion Forecasting Algorithm Based on Pheromone Communication Model (Kurihara, 2013)	To propose a method of congestion forecasting based on multi-agent coordination mechanism.
	Ant Colony Routing algorithm for freeway networks (Cong et al., 2013)	To decrease the travel costs (e.g. time, tolling, fuel consumption, emission) by finding an optimal paths for vehicle routing, and to enhance safety and decrease noise and pollution by distributing the vehicle through the road map.
Hybrid Techniques		
Machine Learning	Solving the Shortest Path Problem in Vehicle Navigation System by Ant Colony Algorithm (Jiang et al., 2007)	To propose a shortest path search method by modifying pheromone update rule and adding learning strategy into ACO.
	The Optimal Routing of Cars in the Car Navigation System by Taking the Combination of Divide and Conquer Method and Ant Colony Algorithm into Consideration (Youse & Zamani, 2013)	To propose an optimal routing method to reduce vehicles travel time by combining Divide and Conquer, ACO and Learning approaches.
	TIME-ANTS: An Innovative Temporal and Spatial Ant-based Vehicular Routing Mechanism (Doolan & Muntean, 2014)	To determine optimal paths for vehicles in both space and time dimensions.

Table 3.3. Ant-based VTRSs considering ACPre (continued)

Ant-based VTRS category	Title of Paper (Ref.)	Objectives
Hybrid Techniques		
Neuro/Fuzzy	Dynamic Fuzzy Logic-Ant Colony System-Based Route Selection System (Salehinejad & Talebi, 2010)	To introduce multi parameter rout selection system utilizing fuzzy logic and neural network for pheromone update and future prediction, respectively.
	Bio-inspired Neuro-Fuzzy Based Dynamic Route Selection to Avoid Traffic Congestion (Abbass et al., 2011)	To propose a bio-inspired neuro-fuzzy based route selection system to avoid vehicle traffic congestion.
	Adapt-Traf: An adaptive multi-agent road traffic management system based on hybrid ant-hierarchical fuzzy model (Kammoun et al., 2014)	To increase the quality of whole road network, especially in the case of traffic congestion conditions, considering real-time traffic data and drivers' travel time.

Figure 3.1 provides a different overview of ant-based VTRS categories and illustrates the relationship of some algorithms with other categories and user preference. For example, approaches proposed by Claes and Holvoet (2011) and Jiang et al. (2007) are in the prediction category because of using link travel time as prediction method. At the same time they investigate ant parameters in their proposed algorithms, hence these approaches can be considered in parameter category as well. The approaches proposed by Salehinejad and Talebi (2010) and Abbass et al. (2011) are also found in the same category and are connected to the user preference, thus indicating that user preference is considered in these algorithms.

<table>
<tr><th></th><th>PARAMETERS</th><th>PREDICTIONS</th><th>SEGMENTATION</th></tr>
<tr><td rowspan="2">Non-User Preferences</td><td rowspan="2">(Hallam et al., 2004)
(Bedi et al., 2007)
(Liu et al., 2007)
(Foroughi et al. 2008)
(Kammoun et al., 2010)
(Zong et al., 2010)
(Nahar & Hashim, 2011)
(Cong et al., 2011)
(Krol & Mrozek, 2011)
(Ge et al., 2011)
(Sur et al., 2012)
(Ghazy et al., 2012)
(Kponyo et al., 2012)</td><td colspan="2">(Tatomir et al., 2009)</td></tr>
<tr><td>(Ando et al., 2006b)
(Kurihara, 2013)
(Yousefi & Zamani, 2013)
(Cong et al., 2013)
(Doolan & Muntean, 2014)</td><td>(Tatomir & Rothkrantz, 2006)
(Narzt et al., 2010)
(Claes & Holvoet, 2012)</td></tr>
<tr><td rowspan="2">User Preferences</td><td colspan="2">(Claes & Holvoet, 2011)
(Jiang et al., 2007)</td><td rowspan="2">-Nil-</td></tr>
<tr><td>(Weyns et al., 2007)
(Ok et al., 2011)
(Boryczka & Bura, 2013)</td><td>(Salehinejad & Talebi, 2010)
(Abbass et al., 2011)
(Kammoun et al., 2014)</td></tr>
</table>

Figure 3.1. Tabular taxonomy of studied algorithms

3.2.2.4. Statistical overview of ant-based VTRS

The reviewed literature has shown that a significant amount of work has been conducted to use ant colonies for designing effective traffic congestion control methods. The statistics, evaluation metrics, and differences of the studied papers are discussed in this section. The results obtained from our statistical investigation are presented in Table 3.4 and Figure 3.2. In Table 3.4 various statistics regarding algorithms evaluation and involved parameters such as size of playground, nodes, links and length are represented. Moreover, statistical information of some of functional parameters such as traffic evaluation and system accuracy are provided. Our investigations show five shortcomings in the studied papers: 1) System overhead and resource management are not considered, 2) Simulation procedures are not properly explained in most cases, 3) Algorithms are not evaluated sufficiently with other functional methods, 4) A suitable approach which considers the non-recurring congestion conditions in its routing procedure has not been proposed yet, and 5) A functional framework for VTRS has not been developed in any of the studies. Therefore, even though the reported protocols have appealing properties and good performance, their actual presentation and evaluation lack true scientific soundness.

Table 3.4. Statistics on methodological approach in the studied papers

Description	Percentage of the papers
Perform empirical evaluation of algorithms	100
Report the number of nodes and links in simulation area	70
Report the size of simulation area	55
Utilize travel time as evaluation metric	55
Report the number of nodes in the network	51
Do not specify simulator name or use a self-made one	48
Compare to other algorithms	44
Report simulation length	33
Report number of iterations	29
Utilize path cost or length as evaluation metric	29
Report the number of vehicles in the simulation	26
Report the relationship between ant parameters and system accuracy	22
Utilize traffic distribution as evaluation metric	22
Report system overhead	7
Utilize velocity of vehicles as evaluation metric	7
Provide a suitable solution for non-recurring congestion conditions	0
Provide a functional and comprehensive framework for VTRS	0

a) Simulation tools for Ant-based VTRS: All of the studied approaches have been assessed by using simulation tools. Choosing a realistic simulation is important for validating the proposed protocol or methodology. In Figure 3.2, we present a chart that illustrates the distribution of various simulation environments used in the studied papers. The investigation results indicate that more than half of the researchers (52%) used development environment applications such as MATLAB and Netlogo (Wilensky, 1999) to simulate the algorithms. Furthermore, 48% of the studied papers used either unknown simulation tools (33%) or self-made simulators (15%). Although traffic simulators (e.g., Sumo) have advanced in recent years, VTRSs still lack specialized simulation tools. The use of self-made simulation tools is infeasible because the production of an accurate and robust simulation environment in computer science requires a considerable amount of effort and long-term collaboration between experts. Almost all of these collaborations lead to well-known simulation tools such as NS2 (Fall & Varadhan, 2014), OPNET (X. Chang, 1999) and OMNET++ (Varga, 2010). Therefore, a self-made simulation environment is unrealistic and is unreliable for evaluating algorithms or methodologies.

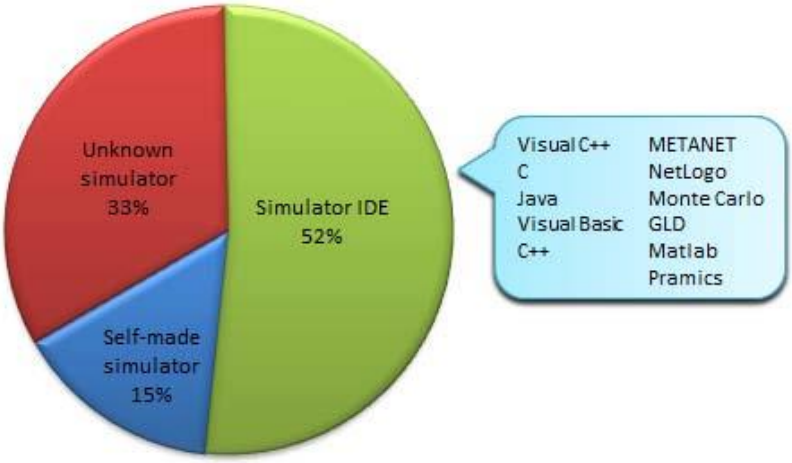

Figure 3.2. Percentage of used simulators for the evaluation of studied algorithms

Tables 3.5 and 3.6 represent evaluation metrics of different algorithms which uses the same IDE as examples. Algorithms which use MATLAB are represented in Table 3.5, while Table 3.6 includes the algorithms which utilize Visual C++ as implementation tool. Moreover, the result of modifying the mentioned parameters in these algorithms is presented in the impact section of each table.

Table 3.5. Comparison of algorithms proposed by Salehinejad and Talebi (2010), and Abbass et al. (2011) using MATLAB

Reference	**Parameters**				**Impact on**
	ρ	Q	α	*Pheromone level*	
(Salehinejad & Talebi, 2010)	0.9	0.9	2	Very weak-very strong	Congestion-distance cost
(Abbass et al., 2011)	0.9	--	0.5	0-5	Route ranking-user preference

Table 3.6. Comparison of algorithms proposed by Liu et al. (2007) and Ge et al. (2011) using Visual C++

Reference	**Parameters**				**Impact on**
	ρ	Q	α	*Number of ants*	
(Liu et al., 2007)	0.5	0.5	0.5	20	Convergence speed
(Ge et al., 2011)	0.7	2	04	50	Route cost

In Tables 3.5 and 3.6, α and β are coefficient of pheromone trail and route cost [Equation (2.1)], ρ is pheromone evaporation [Equation (2.2)] and Q is constant value [Equation (2.3)].

b) Evaluation observation: Although the evaluation and assessment of proposed algorithms are two of the most important and critical parts of research, most studied papers contain the following problems: 1) Insufficient information on the simulation tools and set-up processes, 2) Insufficient statistical information on evaluation metrics (e.g., average travel speed is an important evaluation metric for VTRS; however, only 7% of papers addressed this metric), 3) The obtained results are not discussed or justified in most cases, 4) Only 44% of the papers have compared their results with state-of-the-art algorithms, 5) Only a few datasets and small-scale scenarios are used to evaluate the algorithms.

   For most studies, a single metric is used for evaluation or no evaluation metric is defined at all. Therefore, in the following section, some critical and essential factors that should be considered in future works are discussed. Tables 3.7 and 3.8 indicate the various evaluation metrics used in the studied ant-based VTRS in chronological order. These tables elaborate the impact of desired algorithm on given evaluation metrics. The most used and important evaluation metrics in VTRS are presented in these tables. To mitigate the five mentioned shortcomings in ant-based VTRSs, the following suggestions can be considered.

   1) A proper simulator should be developed to ensure that evaluations are accurate and reliable. Simulation metrics such as the number of ants, simulation duration, size of simulation area, and number of iterations should be defined clearly.
   2) Evaluation metrics should be defined and considered. Some essential evaluation metrics such as travel time, speed, and path length should be evaluated in all algorithms.
   3) Current methodologies should be compared, and authors should publish their simulation codes for accurate comparisons.

4) Algorithms should be defined clearly by using pseudo-codes and should be evaluated based on robustness, scalability, and overhead.

Table 3.7. Comparisons of evaluation metrics between ant-based VTRSs (recent four years)

References	**Evaluation metrics**	**Discussion**
(Kammoun et al., 2014)	Travel time, travel speed, path length	This hybrid probabilistic approach is evaluated by comparing its results with static VTRS (Dijkstra), and dynamic VTRS (predicted travel time) (Xiaoyan Zhang & Rice, 2003). In addition, the influence of the hierarchical fuzzy system on VTRS is examined by evaluating their system with and without it.
(Doolan & Muntean, 2014)	Percentage of vehicles that reached destination	Various approaches (i.e. Dijkstra, mechanism with no rerouting, four types of dynamic rerouting algorithm (Y.-J. Wu & Sung, 2010) and an algorithm from (Sommer, Krul, German, & Dressler, 2010)) are used for performance evaluation. This approach outperforms the others by up to 19% in terms of the mentioned evaluation metric.
(Cong et al., 2013)	Convergent speed, traffic distribution, travel speed, travel time, computation time	Vehicles are routed with and without this approach in order to examine it effect on traffic congestion mitigation. Moreover, this approach is compared with two other dynamic routing approaches (i.e. non-linear optimal control (Kotsialos, Papageorgiou, Mangeas, & Haj-Salem, 2002) and the time-dependent shortest routes method (Tong & Wong, 2000) in order to analyze its performance in terms of various mentioned evaluation metrics.
(Boryczka & Bura, 2013)	Travel time, path length, number of ants	This approach compensates for the inability of dynamic fuzzy logic system (Salehinejad & Talebi, 2010) to support large datasets. This approach also outperforms the ant algorithm and dynamic fuzzy logic system in terms of running time.
(Kurihara, 2013)	Prediction accuracy	This congestion-forecasting algorithm eliminates the need for probe cars and central management servers. This algorithm mainly focuses on predicting changes in traffic density and sudden accidents. However, the need to install several kinds of hardware in various network locations makes the implementation of this method expensive.
(Yousefi & Zamani, 2013)	Travel time, number of ants, path length	The proposed method finds paths by combining the divide and conquers method and ant colony algorithms. This algorithm compared with the Dijkstra and ant colony algorithms in terms of travel time. It exhibits better results than other methods in this comparison.
(Sur et al., 2012)	Number of vehicles, traffic distribution	Various types of vehicles are introduced in this approach to distribute traffic. This method provides better results in avoiding the stagnation of searching criteria compared with traditional ant algorithm.
(Ghazy et al., 2012)	Travel time, number of ants	Based on theoretical investigations, the performance of this algorithm is $O(n^2)$ for road networks with n nodes. The number of ants and travel time are improved by approximately 12% and 3.13%, respectively.
(Kponyo et al., 2012)	Travel speed, waiting time, number of stopped vehicles	In this approach, vehicles' travel speed, waiting time and the number of stopped vehicles are improved compared with the random next-node selection approach.
(Claes & Holvoet, 2012)	Travel time, number of iteration	In this region-based algorithm, every ant agent has to carry (save) additional information to find their target region faster. Hence, although the solution is found quickly, the system uses more resources.

Table 3.7. Comparisons of evaluation metrics between ant-based VTRSs (recent four years) (continued)

References	Evaluation metrics	Discussion
(Ok et al., 2011)	Path length, number of links	Although the best values for a set of parameters (e.g., number of ants and iteration) are introduced to improve ant algorithm's performance, the algorithm is unsuitable for finding the shortest path based on preference.
(Nahar & Hashim, 2011)	Traffic distribution, travel time	This approach is analogous to proposed algorithm by Ok et al. (2011). However this technique does not perform well when the number of agent in the network is less than 100 agents, which leads to high overhead. Travel time can be reduced ranging from 21% to 39% by using this approach.
(Cong et al., 2011)	Traffic distribution, number of ants	This pruning-optimization approach mitigates the dynamic traffic routing problem but does not consider user preference and only optimizes traffic according to the system-optimum concept.
(Krol & Mrozek, 2011)	Travel time, number of stopped vehicles	This method enables the distribution of traffic across multiple servers. Thus, this method can be used to manage traffic flow across very complex road networks with high traffic volume.
(Ge et al., 2011)	Path length, travel time	This Pareto-type method uses a novel network storage structure to make the path-planning algorithm functional. This phase is important for generating the adjacent matrix of the network.
(Abbass et al., 2011)	Prediction accuracy, traffic flow	Route ranking that is based on user preference is provided, and prediction is used to improve the decision-making system. On the basis of the resulting priority-based list, the lowest cost path is selected.
(Claes & Holvoet, 2011)	Travel time, traffic distribution, parameters' relationship, prediction accuracy	A set of ACO parameters that provides a suitable trade-off between exploration and exploitation is found and discussed. Moreover, this algorithm outperforms the A* algorithm in terms of average travel time.

Table 3.8. Comparisons of evaluation metrics between ant-based VTRSs (2010 and below)

References	Evaluation metrics	Discussion
(Narzt et al., 2010)	Travel time, number of stopped vehicles	The pheromone engine designed in this paper is efficient for analyzing traffic information. However, the pheromone trail is not actively monitored because of the cluster-based platform.
(Kammoun et al., 2010)	Travel speed, number of stopped vehicles, travel time	The next node is selected based on two methods: heuristic and probabilistic. The probabilistic method outperforms the heuristic method. Travel speed is increased by an average of 15% compared to static algorithms (e.g., Dijkstra).
(Salehinejad & Talebi, 2010)	Travel time, number of ants, path length, convergent speed	This algorithm is executed locally for each vehicle. Moreover, it outperforms the ant colony system and A* algorithms in terms of convergent speed and average cost.
(Zong et al., 2010)	Number of vehicles	A multi-agent approach that solves the evacuation problem by using vehicle congestion load balancing. This approach outperforms ant colony system in term of evacuation time.
(Tatomir et al., 2009)	Travel time	Four different types of vehicle navigation systems are used to evaluate this algorithm: 1) use of ant-based control algorithms; 2) sending traffic information every 30 min; 3) sending traffic information every 10 min; and 4) using the Dijkstra algorithm. The algorithm that uses ant-based control algorithm has better travel time.
(Foroughi et al., 2008)	Path length, path traffic	Along with path length, path traffic is added to the traffic control system as an optimization metric.

Table 3.8. Comparisons of evaluation metrics between ant-based VTRSs (2010 and below) (continued)

References	Evaluation metrics	Discussion
(Liu et al., 2007)	Convergent speed	A set of parameters are introduced to maintain the operation of the ACO algorithm in a steady state with high convergent speed. The parameter set is related to specific problem space and may not work for other algorithms such as proposed algorithm by Ok et al. (2011) and Nahar and Hashim (2011).
(Weyns et al., 2007)	Traffic distribution	The use of a multi-agent mechanism provides this system with more flexibility for dynamic conditions. However this algorithm does not investigates scalability issues.
(Bedi et al., 2007)	Path length, number of ants	The random function is added to the probability function to increase the exploration rate of agents in the problem space.
(Jiang et al., 2007)	Path length, number of ants, convergent speed, number of iteration	Three different transition rules based on density, distance, and angle between two paths are used in this algorithm. The angle-based transition rule outperforms the other two rules in terms of path length. An agent is called a ‘dead ant’ if it cannot find a feasible path from source to destination.
(Tatomir & Rothkrantz, 2006)	Travel time	This approach uses hierarchical ant-based control algorithm to provide scalability for its routing system. Travel time is reduced by quickly reacting to new changes.
(Ando, Masutani, et al., 2006)	Travel time, prediction accuracy, parameters’ relationship	Two types of vehicles are used in this approach: general and commercial (e.g., buses or taxis). Thus, different pheromone types are assigned to different vehicle types. High prediction accuracy is obtained by using pheromone prediction method.
(Hallam et al., 2004)	Path length, number of ants	Path length This parameter-based method investigates the best available values to be set for ant algorithm. However, this method cannot support fairness.

c) Comparison of studied approaches: The similarities and differences between various studied ant-based VTRSs along with their probability functions are discussed. This information can provide insights on the studies conducted regarding ant-based VTRS for designing and forming future frameworks. Some shortcomings, such as the lack of novel and effective probability functions, can be recognized through the following discussions.

Features such as data gathering, ant agent, new pheromone, and new ant type should also be considered in ant-based VTRSs. Table 3.9 discusses these features in detail. Data gathering is an important phase that uses either historical information, real-time information, or both. Moreover, ant agents involved in VTRSs can be virtual or real (vehicle). To enhance the performance of the ant-based algorithm, in some studies, the basic pheromone type is modified and a new type of pheromone is proposed, such as stench and colored pheromones. New ant types, such as check and colored ants, are also proposed and used in path finding process.

Table 3.9. Features and specifications of various ant-based VTRSs

Ant-based VTRSs	Reference	Data gathering	Ant agent	New pheromone	New ant
Ant Colony Parameters					
Variable-based	(Nahar & Hashim, 2011)	Historical	Real	No	No
	(Ok et al., 2011)	Historical	Virtual	No	No
	(Liu et al., 2007)	Historical	Virtual	No	No
Step-based					
Initialization	(Boryczka & Bura, 2013)	Real-time	Virtual	Yes	No
	(Ghazy et al., 2012)	Historical	Virtual	No	Yes
	(Sur et al., 2012)	Historical	Virtual	Yes	Yes
	(Cong et al., 2011)	Real-time	Real	Yes	Yes
	(Krol & Mrozek, 2011)	Historical	Real	No	No
	(Zong et al., 2010)	Real-time	Real	Yes	Yes
	(Kammoun et al., 2010)	Real-time	Real	No	Yes
	(Foroughi et al., 2008)	Real-time	Virtual	No	No
	(Weyns et al., 2007)	Real-time	Virtual	No	Yes
Probability Function	(Kponyo et al., 2012)	Real-time	Real	No	No
	(Ge et al., 2011)	Historical	Virtual	No	No
	(Bedi et al., 2007)	Real-time	Virtual	No	No
	(Hallam et al., 2004)	Hybrid	Virtual	Yes	Yes
Ant Colony Segmentation					
	(Claes & Holvoet, 2012)	Hybrid	Real	Yes	No
	(Narzt et al., 2010)	Real-time	Real	Yes	No
	(Bogdan Tatomir & Rothkrantz, 2006)	Real-time	Virtual	No	No
Ant Colony Prediction					
	(Cong et al., 2013)	Real-time	Virtual	Yes	Yes
	(Kurihara, 2013)	Real-time	Real	Yes	No
	(Claes & Holvoet, 2011)	Historical	Virtual	No	No
	(Bogdan Tatomir et al., 2009)	Hybrid	Virtual	Yes	Yes
	(Ando, Masutani, et al., 2006)	Real-time	Real	Yes	No
Hybrid technique	(Kammoun et al., 2014)	Real-time	Virtual	No	No
	(Doolan & Muntean, 2014)	Hybrid	Virtual	No	No
	(Yousefi & Zamani, 2013)	Historical	Virtual	Yes	No
	(Abbass et al., 2011)	Hybrid	Virtual	Yes	No
	(Salehinejad & Talebi, 2010)	Hybrid	Virtual	Yes	No
	(Jiang et al., 2007)	Hybrid	Virtual	Yes	Yes

Ants select the next hop to reach their destinations by using the probability function. Various probability functions in studied methodologies and their parameter descriptions are presented in Table 3.10. Most probability functions are similar to the basic probability function (represented in the first row of the Table 3.10); the only difference is minor modifications such as the addition or omission of parameters from the basic probability function.

Table 3.10. Different probability functions used in ant-based VTRSs

Reference	Probability function	Description
(Marco Dorigo, 1992)	$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{h \in tabu_k}[\tau_{ih}(t)]^{\alpha}[\eta_{ih}(t)]^{\beta}}$ (basic probability function)	Intensities α and β are used to stress the importance of trail $\tau_{ij}(t)$ and route cost $\eta_{ij}(t)$. *tabu*$_k$ is the set of visited nodes for ant *k*.
(Hallam et al., 2004)	$P_{ij}^{k}(t) = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[nol_{ij}]^{\lambda}[1/noc_{ij}]^{\delta}}{\sum_{l \in Allowed_k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[nol_{ij}]^{\lambda}[1/noc_{ij}]^{\delta}}$	Intensities α, β, λ and δ are the relative importance that can be used stress on the importance of the trail $[\tau_{ij}]$, visibility $[\eta_{ij}]$, number of lanes $[nol_{ij}]$, and number of cars $[noc_{ij}]$, respectively.
(Bedi et al., 2007)	$P_{ij}^{k}(t) = \frac{(\tau_{ij}(t))^{\alpha}\eta_{ij}{}^{\beta}}{\sum_{l \in Allowed_k}(\tau_{ij}(t))^{\alpha}\eta_{ij}{}^{\beta}}$ + Random function	$\tau_{ij}(t)$ is the intensity of pheromone trail on edge *(i, j)* at time *t*, η_{ij} (visibility factor) = $1/d_{ij}$ (d_{ij} is the distance between nodes *i* and *j*). α and β are the parameters that control the relative importance of the pheromone trail vs. visibility.
(Kammoun et al., 2010)	$P_{itinerary}^{i} = \frac{\left(Q_{itinerary}^{i}\right)^{\alpha}\left(\frac{W_{shortest-itinerary}}{W_{itinerary}^{i}}\right)^{\beta}}{\sum_{j=1}\left(Q_{itinerary}^{j}\right)^{\alpha}\left(\frac{W_{shortest-itinerary}}{W_{itinerary}^{j}}\right)^{\beta}}$	$Q_{itinerary}^{i}$ is the quality of itinerary *i*. $W_{itinerary}^{i}$ is the itinerary weight representing the itinerary length, and $W_{shortest-itinerary}^{i}$ is the length of the shortest possible itinerary. α and β represent the itinerary intensities.
(Salehinejad & Talebi, 2010)	$P_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\prod_{l \in parameters}\xi_{ijl}^{-\alpha_l}}{\sum_{h \notin tabu_k}\tau_{ih}^{\alpha}\prod_{l \in parameters}\xi_{ihl}^{-\alpha_l}}$	τ_{ij} is the direct route pheromone intensity from junction *i* to *j*. Parameter α the importance of τ_{ij}. The *tabu*$_k$ is the set of visited nodes. The cost function of each parameter *l* is $\xi_{ijl}^{-\alpha_l}$, where $1 \leq \xi_{ijl} \leq 10$ and the significance of each *l* is adjustable using α_l for all parameters
(Claes & Holvoet, 2011)	$P_{ij} = \frac{(1-\Upsilon)\tau_{ij}^{\alpha}.\Upsilon\eta_{ij}^{\beta}}{\sum_{(i,n)\in S}(1-\Upsilon)\tau_{in}^{\alpha}.\Upsilon\eta_{in}^{\beta}}$	Intensities α, β and ϒ are the relative importance that can be used to stress the importance of the trail τ_{ij} and visibility η_{ij}.
(Ok et al., 2011)	$P_{ij}^{k}(t) = \frac{1/([\tau_{ij}(t)][\eta_{ij}(t)]^{\beta})}{\sum_{u \in N_i^k(t)}1/([\tau_{iu}(t)][\eta_{iu}(t)]^{\beta})}$	$N_i^k(t)$ is the set of candidate nodes connected to node *i*. $\tau_{ij}(t)$ and $\eta_{ij}(t)$ represent the pheromone trail and a heuristic function that is defined as the inverse of the distance between node *i* and *j*, respectively. Parameter β is a positive constant and is used to amplify the influence of the heuristic function.
(Cong et al., 2011)	$P_c\{j\|i\} = \frac{(\max\{\tau_{min}.\tau_{ij}\})^{\alpha}}{\sum_{l \in N_{i,c}}(\max\{\tau_{min}.\tau_{il}\})^{\alpha}}$ $\forall j \in N_{i,c}$	τ_{ij} is the pheromone level previously deposited by the ants on link *(i, j)*. Parameter α determines the relative importance of τ_{ij}. The feasible neighborhood $N_{i,c}$ of ant *c* at node *i* is the set of nodes that are connected to *i* and the unvisited nodes of ant *c* in the current iteration. τ_{min} is the minimum pheromone level on each link, and guarantees the low pheromone level, and thus preventing the denominator of the formula from becoming *zero*.

Table 3.10. Different probability functions used in ant-based VTRSs (continued)

Reference	Reference	Reference
(Claes & Holvoet, 2012)	$P_{ij}^{c} = (1-\lambda)P_{ij} + \lambda\sigma_{ij}(r)$	r is the region the ant is attempting to reach, P_{ij} is the probability of choosing edge (i, j) from (20), and λ is a weighing factor between the vehicle specific and region-specific information. $\sigma_{ij}(r)$ is the region-specific pheromones for every region *r* reachable through that edge.

3.2.2.5. Characteristics of ant-based VTRS

On the basis of the aforementioned discussions, the characteristics of ant-based VTRSs are discussed in this section. Related literature has classified these characteristics in different ways. However, the deterministic/stochastic (as technique), reactive/predictive (as strategy), flat/hierarchical (as topology), centralized/decentralized (as architecture), and loop free/load balancing characteristics of ant-based VTRSs are considered and discussed in this subsection. These characteristics are summarized in Table 3.11. According to Table 3.11, most of the VTRSs use predictive and flat mechanisms. In addition to these characteristics, dynamicity and optimality should always be considered in VTRS. Considering the nature of these systems, all algorithms and techniques introduced in this area should always be dynamic and optimal.

1) Technique: The first characteristic of ant-based VTRSs is related to the treatment of travel cost. This value can be computed in a deterministic or stochastic manner (Schmitt & Jula, 2006). Deterministic systems assign pre-defined and deterministic vales for links (roads) and disregard the dynamic and random nature of vehicle congestion. By contrast, stochastic systems consider the traffic condition of roads (links) and assign different values to links according to historical or/and real-time traffic information. Although stochastic systems require high computational capacity to process large amounts of vehicle traffic data, stochastic systems are not vulnerable to the random nature of vehicle congestion. Stochastic approaches outperform deterministic approaches in solving dynamic problems.

Table 3.11. Characteristics of studied ant-based VTRSs

References	Characteristics					
	Technique	*Strategy*	*Topology*	*Architecture*	*Loop free*	*Traffic load balancing*
(Kammoun et al., 2014)	Stochastic	Predictive	Hierarchical	Decentralized	Yes	Yes
(Doolan & Muntean, 2014)	Stochastic	Predictive	Flat	Centralized	No	Yes
(Cong et al., 2013)	Stochastic	Predictive	Flat	Centralized	Yes	Yes
(Kurihara, 2013)	Stochastic	Predictive	Hierarchical	Decentralized	No	No
(Yousefi & Zamani, 2013)	Deterministic	Predictive	Hierarchical	Decentralized	No	Yes
(Boryczka & Bura, 2013)	Stochastic	Predictive	Flat	Decentralized	Yes	No
(Claes & Holvoet, 2012)	Stochastic	Predictive	Hierarchical	Centralized	No	No
(Sur et al., 2012)	Stochastic	Predictive	Flat	Decentralized	Yes	Yes
(Ghazy et al., 2012)	Stochastic	Predictive	Flat	Decentralized	Yes	No
(Kponyo et al., 2012)	Stochastic	Predictive	Flat	Decentralized	No	Yes
(Ge et al., 2011)	Deterministic	Predictive	Flat	Decentralized	Yes	No
(Ok et al., 2011)	Deterministic	Predictive	Flat	Decentralized	Yes	No
(Nahar & Hashim, 2011)	Deterministic	Predictive	Flat	Decentralized	Yes	Yes
(Cong et al., 2011)	Stochastic	Reactive	Hierarchical	Decentralized	No	Yes
(Krol & Mrozek, 2011)	Stochastic	Predictive	Flat	Decentralized	Yes	Yes
(Abbass et al., 2011)	Stochastic	Predictive	Flat	Decentralized	Yes	No
(Claes & Holvoet, 2011)	Stochastic	Predictive	Flat	Centralized	Yes	Yes
(Kammoun et al., 2010)	Stochastic	Predictive	Flat	Decentralized	Yes	Yes
(Zong et al., 2010)	Stochastic	Reactive	Flat	Decentralized	No	Yes
(Salehinejad & Talebi, 2010)	Stochastic	Predictive	Flat	Hybrid	Yes	No
(Narzt et al., 2010)	Stochastic	Predictive	Hierarchical	Decentralized	Yes	No
(Tatomir et al., 2009)	Deterministic	Predictive	Hierarchical	Decentralized	Yes	No
(Foroughi et al., 2008)	Stochastic	Predictive	Flat	Decentralized	Yes	No
(Jiang et al., 2007)	Stochastic	Predictive	Flat	Decentralized	Yes	No
(Liu et al., 2007)	Deterministic	Predictive	Flat	Decentralized	Yes	No
(Weyns et al., 2007)	Stochastic	Predictive	Flat	Hybrid	No	Yes
(Bedi et al., 2007)	Stochastic	Predictive	Flat	Decentralized	Yes	Yes
(Tatomir & Rothkrantz, 2006)	Stochastic	Predictive	Hierarchical	Centralized	Yes	No
(Ando et al., 2006)	Stochastic	Reactive	Flat	Hybrid	No	Yes
(Hallam et al., 2004)	Stochastic	Predictive	Flat	Decentralized	Yes	No

2) Strategy: Reactive ant-based VTRSs use current traffic information and disregard the future conditions of vehicle congestion, whereas predictive ant-based VTRSs use prediction models and historical information to estimate future congestion conditions. Low complexity and robustness are the main advantages of reactive and Predictive ant-based VTRSs, respectively. Thus, one of these characteristics can be selected based on the goal of the system (i.e., low complexity or robustness).

3) Topology: The entire routing map is considered one level in flat ant-based VTRSs, and routing may occur between two arbitrary nodes as the source and destination. However, in hierarchical ant-based VTRSs, the routing map includes either different levels or different regions (cluster or zone) for route decisions. Each level or region

has one or more special nodes (i.e., cluster head and border node). Hierarchical ant-based VTRSs use these nodes to route vehicles between different regions, and vehicles use these nodes to enter new regions. Although both of these systems can be useful for vehicle routing, hierarchical ant-based VTRSs manage dynamic changes better than flat VTRSs. Thus, hierarchical ant-based VTRSs outperform flat systems in terms of vehicle congestion control.

4) Architecture: Centralized ant-based VTRSs use one node as a server/base station for discovering and maintaining routes; this node broadcasts routing information among vehicles. Thus, this node sustains system operations and its failure leads to the failure of the whole system. By contrast, in decentralized ant-based VTRSs, each node gathers and builds routing table for its own use (Di Caro & Dorigo, 2011; Schmitt & Jula, 2006). Although these systems are adaptable to the dynamic nature of vehicular networks, vehicles (nodes) require powerful processing units to execute tasks. Reliability and a superior vision of the routing process are the main advantages of centralized ant-based VTRSs (Schmitt & Jula, 2006). However, the drawbacks of centralized ant-based VTRSs are delay and scalability, which can be solved by decentralized ant-based VTRSs. In order to take advantages of both centralized and decentralized architectures, distributed centralized architecture is introduced by using map segmentation and layering model. To find the best path for each area of the map a management server is assigned to each segment and border nodes are responsible to perform the routing among the segments. More information about map segmentation and layering model and distributed centralized architecture can be found in Section 4.3.1.1.

5) Loop free: Data packets are used as ant agents to find the optimal paths. If packets use a path that has no cycles to traverse the path between the source and destination, the ant-based VTRS is called ‘loop free’. To add this feature to a system, a

monitoring mechanism should be built to avoid any possible loop or cycle. Getting into loops has several negative network effects, such as throughput degradation and increased delays. Looping also wastes bandwidth and energy resources.

6) Traffic load balancing: Ants select paths at each intermediate node according to the distribution of deposited pheromones at each node. If a developed ant-based VTRS uses a pattern that efficiently splits traffic between different paths from the source to the destination, such an approach method can be claimed to apply load balancing successfully.

3.3. Conclusion

This chapter discussed the existing studies related to vehicle traffic routing and congestion mitigation systems and approaches. Accordingly, static algorithms are used in initial systems in order to find the shortest paths which is not a proper solution. Over the years, dynamic routing algorithms are proposed to overcome the main drawback of static routing algorithms which is lack of considering real-time traffic information in their routing mechanism. Among dynamic VTRSs, bio-inspired-based VTRSs, especially those are based on ant, bee, genetic and PSO algorithm are reported as promising approaches for solving multi-criteria shortest path problem. It is worth noting that ant-based VTRSs are reported as the most suitable approach for solving vehicle traffic routing and congestion issues through the literature due to its better adaptability as well as lower processing time. In order to get more details about these approaches, a novel taxonomy which divides ant-based VTRSs into three main classes, namely ACPs, ACS and ACPre, is proposed and discussed in this chapter. Moreover, based on statistical overview of ant-based VTRSs, there are several shortcomings in these systems including specialized simulation tools, novel and effective probability function, considering non-recurring congestion conditions in vehicle routing procedure and a generic framework. The last two shortcomings are the concerns of this thesis and are discussed in Chapter 4. Hence, the major characteristics of VTRSs which should be considered in designing vehicle traffic routing and congestion mitigation framework are discussed in Chapter 3.

CHAPTER 4: DEVELOPED FRAMEWORK DESIGN

In this chapter we detail the design aspects of developed framework to achieve our main goal which is designing an ant-based framework for avoiding vehicle traffic congestion by using VANETs. In other words, vehicles are routed from their origins to destinations in a way to avoid congestion occurrence instead of recovering from it. Later we will evaluate and validate this framework through the simulation environment. This chapter consists of three main topics, namely methodology, design challenges of ant-based VTRS framework and developed ant-based VTRS framework. In Section 4.1, the general steps of our research methodology and utilized research method are outlined. Section 4.2 discusses the challenges that should be considered for designing an ant-based VTRS framework. A description of our developed framework, called AVCAF, is given in Section 4.3. Section 4.4 gives an overview of AVCAF implementation in real world. Finally, Section 4.5 concludes this chapter.

4.1. Methodology

Our methodology in this thesis includes four main steps as follows:

a) Literature review: In this step, a comprehensive and critical overview and analysis of recent developments and researches related to vehicle traffic congestion problem, especially those concentrating on ant-based algorithms for solving this problem, are carried out. VTRSs that utilize ant-based algorithms are classified into three classes based on our proposed taxonomy, namely ACPs, ACPre and ACS. In addition, pros and cons of these approaches are specified in order to find the existing gaps in VTRSs, especially in vehicle traffic congestion field.

b) Modeling: After investigating the existing related works and literature and considering their analyzed results, the required characteristics and design challenges of ant-based VTRSs are obtained and categorized in Section 3.2.2.5 and Section 4.2, respectively. The newly developed ant-based framework (i.e. AVCAF), which is the first framework in this research area, is proposed taking into account the major drawbacks of the existing approaches such as scalability, single path suggestion, non-recurring congestion consideration, congestion avoidance instead of recovering from it and high usage rate. In this step, all the details of the new framework are discussed, including the assumption, architectural perspective, its various modules and phases.

c) Developing: Our proposed framework utilizes vehicular networks for real-time data gathering and also for distributing route guidance information among vehicles. Due to the unique characteristics of these networks such as lack of central coordination, dynamic topology, error prone shared radio channel, limited resource availability, hidden terminal problem and insecure medium, experimentation and performance evaluation of our developed framework can be achieved via simulation tools. Real test-beds construction for any vehicular networks' scenario is an expensive or in some cases impossible task if metrics such as testing area, mobility and number of vehicles are taken into account. Besides, most experiments are not repeatable and require high cost and efforts. Simulation tools can be used to overcome these problems (Martinez, Toh, Cano, Calafate, & Manzoni, 2011b and Schilling, 2005). As a result, after designing and modeling the new framework and it details, the NS-2 (McCanne, Floyd, Fall, & Varadhan, 1997) is utilized for developing our framework. Besides, SUMO (Krajzewicz, Erdmann, Behrisch, & Bieker, 2012) is the most widely used open-source and time discrete microscopic road traffic simulation package available. It is used to generate the vehicle traffic and movement patterns in this thesis. Moreover, TraNSLite, which is a Graphical User Interface (GUI) tool for generating realistic mobility traces for simulating vehicular networks in NS-2, is used to convert the generated traffic scenario into a usable format for NS-2.33. The output of TraNSLite is a TCL file which is used as the traffic pattern for NS-2. These simulation tools are discussed in more details in section 5.1.

d) Testing and evaluation: In this step, extensive and various simulation runs and tests are carried out to evaluate and validate the performance of our approach compared with other existing ant-based and other bio-inspired approaches. In addition, best values for newly developed ant-based framework variables are determined through the simulation runs.

   Different simulation scenarios with various vehicle densities, velocities, city maps with different sizes, and weather and accident conditions are considered in order to have comprehensive comparison between our approach and existing solutions. Average travel time, distance and speed, system usage rate, process time and convergence speed are used as comparison and performance evaluation metrics.

In addition, Research methods are classified based on different considerations, namely: the research purpose (basic or applied), the nature of the principal data (quantitative or qualitative), or the kind analysis conducted (analytical or descriptive). Quantitative and qualitative methods are the most common research methods in computer science field. Hence, their definition and characteristics are discussed in the following section.

4.1.1. Quantitative vs. Qualitative

The method of research which is known as quantitative, gather the numerical data. The analysis of such data will lead to find results which are comparable with the results of other studies on the same topic. For the analysis of data in quantitative research, numerous analytical tools are employed such as statistical methods. On the other hand, the data compiled in the qualitative method is in form of objects, words or pictures. Qualitative research is mainly appropriate for searching on the topics that the information available is scarce and the design should be explorative. A research methodology example for qualitative research is an ornithologist observing the behaviors of a newly identified species of birds. For the earlier stages of the research, qualitative method is ideal while the quantitative method is highly recommended for the latter part of the research project. It is believed that the picture provided in the quantitative method is clearer for the researcher compared to the qualitative method.

In qualitative research, the researcher acts as the primary data gathering tool where he employs different strategies regarding data gathering such as structured interviews, non-structured interviews, documentary analysis, narratives, observation and so on. In the quantitative research, however, employs instruments such as surveys, questionnaires, tests, and other tools to gather measurable and numerical data. The presentation of data in a qualitative research can be in the form of words, which are collected from interviews. It can also be in the form of images from videos and also other objects such as artifacts. Figures or graphs and statistics data in a table form can be regarded as the examples of quantitative research data presentation. Therefore, the research method which has been employed in this thesis can be considered as the quantitative method due to the following reasons:

1. Because it has been planned to implement and evaluate the developed framework in NS-2 which will be discussed later and it produces data in the form of numbers which can be analyze using statistical software.
2. Representation of analyzed data will appear in the form of tables and figures.
3. Analyzed data from the new mechanism will be compared with the previous mechanisms.

After discussing about our methodology, design challenges for Ant-based VTRSs are explained in the following section.

4.2. Design challenges of Ant-based Vehicle Traffic Routing Systems

In addition to characteristics of VTRS which are discussed in Section 3.2.2.5, there are some other features that distinguish the ant-based VTRS from traditional vehicle guidance systems. These features are initialization, pheromone deposition, multi-agent system, pareto archive and prediction. These features are the main ACO attributes that affect VTRSs directly in the data gathering, path finding and path suggestion procedures. Proper values should be assigned to these attributes' variables in order to achieve optimal solutions. VTRSs use real time or historical information to find the preferred path to a particular destination. VTRSs can be implemented by using different factors that are important for the user, such as toll-free path, accident-free path, and shortest path. VTRS has to prepare the required information in order to propose optimal and proper paths for drivers. The majority of VTRSs require the deployment of sensor-equipped vehicles. Hence, the scalability of the VTRS is also a major concern. VTRSs also require self-organization and route optimization mechanisms to guide vehicles.

a) *Initialization:* The process of constructing solutions for traffic control in an ACO system can be considered a graph construction wherein each edge of the graph represents the possible path of an ant. The first and initial challenge in designing traffic control algorithms is the initialization of default information such as edge weight and value. Ant movements are guided by
   1) Heuristic information (η) that represents prior data on the problem and,
   2) Pheromone trails (τ) that encodes information on the ant colony search process, which is continuously updated.

These values are used by ants in conducting probabilistic decisions for the next visited nodes.

b) *Pheromone deposition/update:* Ants conduct pheromone deposition and update procedures to update the pheromone metric. This procedure can be applied when multiple matrices are used. For this case, one can select a set of best solutions (i.e., shortest path) to update the pheromone metric. Another method to update the pheromone metrics is to gather and store non-dominated solutions in an auxiliary function (Ahangarikiasari, Saraji, & Torabi, 2013). On one hand, individuals can update a specific pheromone matrix or some pheromone matrices. On the other hand, all ants are allowed to update the pheromone. The implementation and optimization of a solution is an important attribute in designing ACO algorithms.

c) *Multi-agent system:* The multi-agent colony approach requires a number of ants to construct a colony. Agents individually construct solutions by using their own pheromone and heuristic information and conduct search procedures on particular areas of the problem graph. The agents are able to collaborate with each other by exchanging information, sharing solutions, and updating pheromone values. Therefore, the solutions generated by certain agents affect the pheromone information of other agents. In other words, although each ant agent explores the problem graph and constructs a solution individually, its found solution will be affected by other ants and also will affect the found solution by the other ants.

d) *Pareto archive (Solution Set):* One of the problems which is usually faced when dealing with VTRSs is finding the optimal route between two selected points on the given map, taking into consideration different objectives such as distance, traffic load, road width, risk of collision, quality and number of intersections. It does not mean that this route should be the shortest one. It should be optimal in term of user preferences or objectives. The problem is known from the literature as multi-objective shortest path problem (Horoba, 2010; Ulungu & Teghem, 1991), and it is proven to be NP-complete (Hansen, 1980). For multi-objective combinatorial problems a single solution is very rarely be able to minimize (or maximize) all objectives, but rather there will be a set of compromise solutions. These solutions are called efficient non-dominated ones and are also referred to as Pareto optimal set. The solution in the Pareto set is used by multi-objective ACO algorithms to update the pheromone information (García-Martínez, Cordón, & Herrera, 2007). The algorithm indicates how the Pareto set is stored and used during implementation. A

Pareto archive can be established in Pareto-based algorithms by using two approaches, namely, offline and online storage. These techniques allow the pheromone matrix or matrices to affect the state of the non-dominated set at any time.

e) *Prediction and detection:* Traffic congestion occurs when the number of vehicles exceeds the capacity of the road. As a result, in half of the cases, traffic condition estimation from historical data is accurate enough for vehicle routing due to repeatable nature of traffic condition which is called recurring (i.e. normal or expected) traffic congestion. But, recurring congestion only includes 50% of congestion conditions on the road and the other 50% is related to non-recurring (i.e. non-routine or unexpected) events such as accidents, vehicle breakdowns and weather conditions (Coifman & Mallika, 2007). Current traffic data can be used for considering these events in vehicle congestion problem. The way of utilizing gathered data to predict and detect the future events on the roads and the topology of vehicles is a critical issue and should be considered in the design of VTRSs.

f) *Scalability:* Scalability is defined as the ability of dealing with the addition of nodes or vehicles without suffering a noticeable loss in performance or increase in administrative complexity (Neuman, 1994). VTRSs are expected to remain operational for an infinite period in a wide geographical area. New vehicles may enter and leave the communication range of current vehicles. Thus, the number of vehicles in communication ranges changes continuously and often becomes unpredictable. An effective vehicle traffic control system should be able to cope with the changes and challenges that originate from vehicular and wireless communication networks.

g) *Self-organization and route optimization:* Vehicle drivers are the main users of traffic control systems and need to form an ad-hoc network to send and receive real-time traffic information. Self-organization should always be considered in the design of VTRSs. Centralized control is not suitable for vehicular networks because of the high-speed changes in topology (Sjöberg Bilstrup, 2009). To be effective, a VTRS must be resilient to dynamic and unpredictable variations. Therefore, VTRSs should find the shortest path and optimize such a path continuously to overcome path changes and consider various objectives. Thus, link weights should be changed by using real-time information to find the optimal route. Essential services should also be available for the long-term use of a decentralized system.

4.3. Ant-based Vehicle Congestion Avoidance Framework (AVCAF)

This section discusses about our proposed and developed framework (i.e. AVCAF). AVCAF is depicted in Figure 4.1 and aims to reduce vehicle traffic congestion problem by using its various phases. 1) Initialization, 2) Optimal path finding, and 3) Optimal path suggestion are the three phases involved in AVCAF. To the best of our knowledge, this framework is the first framework that is related to ant-based vehicle traffic routing systems. The initialization phase consists of data gathering and map preparation. In the data gathering phase several tools such as camera, wireless sensors and inductive loop detection are utilized in order to collect required information for the initial step of AVCAF. Each of these three tools operates as follows: By using video feeds from the cameras, the built-in software harvests information from that video, then gathered information such as vehicle volume and average velocity are fed into the fuzzy system. Moreover, wireless sensors are deployed by the road intersection to detect vehicles. These sensors send the collected data to the intersection control agent. Then it processes the data and dynamically controls the vehicle traffic. Additionally, inductive loop detection can be placed in a roadbed to detect vehicles by measuring the vehicle’s magnetic field. The output of the gathered data rather than the gathering process is important in ant-based VTRS. The main functionality of ant algorithm in AVCAF exists in Phases 2 and 3, where an optimal path should be selected and suggested based on gathered data. In Phase 2, values or weights are assigned to the roads on the map according to the output data of Phase 1. In addition to link value assignment, these gathered data, especially vehicles speed, density and travel time, are used for non-recurring congestion detection. Based on our investigation, these three attributes are affected via non-recurring or unpredictable events such as accident, working zones, weather conditions. As a result, these attributes should be considered in any vehicle congestion avoidance mechanism. Policies are assigned to AVCAF based on the environmental variables. For instance, different policies can be used for different times in a day (i.e. the vehicle traffic congestion condition is different during the daytime versus in the night-time). In Phase 3, the optimal path is suggested by using the obtained information from previous phase and used pheromone update rule. AVCAF is discussed in detail in the subsequent sections of this thesis.

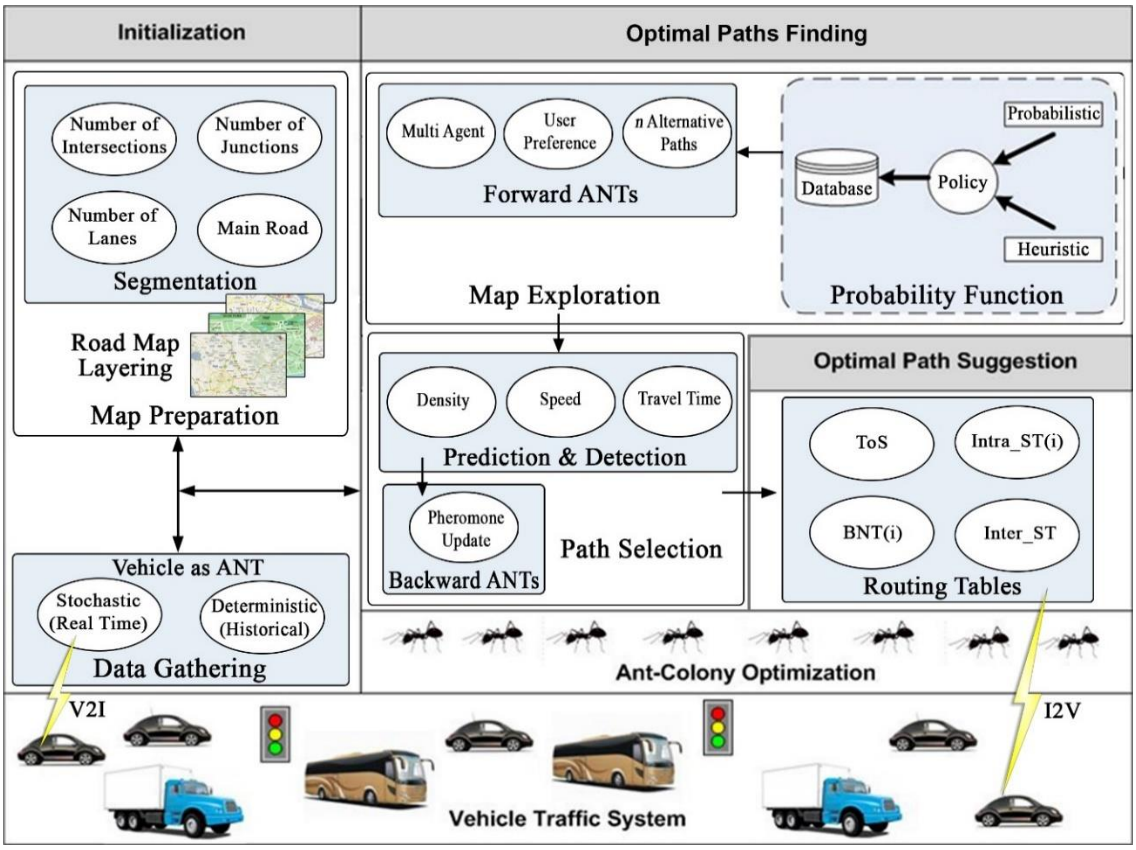

Figure 4.1. Schematic view of AVCAF

All of the discussed characteristics of VTRSs in section 3.2.2.5 and design challenges of ant-based VTRSs in section 4.2 are considered in AVCAF by usage of ant-based algorithm. Table 4.1 gives an overview about characteristics and design challenges of VTRSs and the way of considering them in AVCAF.

Table 4.1. The way of considering various characteristics and design challenges of VTRSs in AVCAF

	Characteristic	**The way of consideration in AVCAF**
Design challenges of ant-based VTRSs	Initialization	It includes map preparation and data gathering which are considered in the first phase of AVCAF.
	Pheromone update	It is used for next node (e.g. junction or intersection) selection and is considered in one of the steps of ant-based algorithm in AVCAF (section 4.3.3.1).
	Multi-agent system	Ant algorithm is a multi-agent system in its nature. As a result this design challenge is considered in AVCAF since it used ant-based algorithm.
	Pareto archive (Solution set)	Each individual ant explores the problem graph, finds and stores an optimal solution in its memory. These optimal solutions form Pareto archive and is used by AVCAF to update the links' pheromone intensity.
	Prediction and Detection	It includes vehicles speed, density and travel time, and is used for both recurring and non-recurring congestion detection. This issue is considered in the second phase of AVCAF for link value assignment (section 4.3.2.1).
	Scalability	As the number of nodes on the map increases, the number of ant agents also increases, which means that ant-based algorithms suffer from poor scalability (Z. Xu, Hou, & Sun, 2003). This drawback is considered in the first phase of AVCAF, especially in map preparation. It can be solved via segmentation technique (section 4.3.1.1).
	Self-organization and Route optimization	Vehicular ad hoc networks and distributed centralized servers are utilized in AVCAF in order to consider self-organization characteristic. In the case of route optimization, ant agents are used to find the optimal paths and optimize found paths considering various future and upcoming conditions on the roads.
Characteristics of VTRSs	Deterministic/ Stochastic	Both of these characteristics are considered in the first phase of AVCAF, especially in data gathering phase which takes place via VANET (section 4.3.2.1).
	Reactive/ Predictive	Predictive characteristic is chosen for our developed frameworks due to dynamic nature of vehicular traffic and networks. Upcoming topology and events should be predicted and considered in vehicle routing procedure in order to achieve our main goal (i.e. vehicle congestion avoidance). Predicted travel speed of vehicles is used in AVCAF for this purpose.
	Flat/ Hierarchical	Hierarchical characteristic is selected in the map preparation phase of AVCAF in order to reduce the system complexity and overcome to dynamic nature of vehicular environments (section 4.3.1.1).
	Centralized/ Decentralized	One of the main requirements of VTRSs, using decentralized processing system, is achievable through the segmentation sub-module of map preparation in AVCAF. It means that several centralized servers (navigators) are distributed throughout the segmented road map (i.e. one server for each segment), namely distributed centralized servers, to guide the vehicles to their destinations (section 4.3.1.1).
	Loop free	Stopping procedure is defined in the third phase of AVCAF in order to consider this characteristic in our developed framework (section 4.3.3.2).
	Traffic load Balancing	This characteristic is the main goal of current thesis and AVCAF is developed and designed in a way to achieve this goal. The obtained results prove our claim.

Before explaining the various phases of AVCAF in more details, we discuss the technical architecture that AVCAF uses and whether it is centralized or decentralized. In a decentralized architecture, route finding and computation takes place for each vehicle individually using the on board processor and memory of that vehicle. It is ideal if vehicles receive traffic information through wireless communication (e.g. V2V or V2I) and include road maps and GPS. In a centralized architecture, route finding and computation takes place through a central server in response to requests from drivers. In this architecture, the central server has access to the historical or real-time traffic information database and computes routing algorithms based on this information. More information about centralized or decentralized architectures and their pros and cons can be found in the study by Suson (2010).

AVCAF is designed for decentralized architecture; however the architecture is not yet available to be used, since all vehicles are not equipped with on-board units and wireless transceivers. AVCAF can be configured to be distributed over several on-board vehicle navigation systems, but at this moment, it can only be executed in a centralized manner. Therefore, AVCAF is part of a group of distributed centralized servers (section 4.4.1 for more details) which provides route advice to the drivers who are equipped with transceiver devices (e.g. mobile phones, PDAs, Tablets). Our model also assumes that the system can detect the position of the vehicles via GPS. AVCAF is expected to guide vehicles through the least congested shortest paths to their destinations. As was discussed previously, initialization, optimal path finding and optimal path suggestion are the main phases of AVCAF (section 4.3). Each of these phases is discussed in the following subsections.

4.3.1. Initialization

The initialization phase consists of two modules, namely data gathering and map preparation. It includes the initial steps of our AVCAF framework where the required real-time traffic data for vehicles routing are gathered and the road map is prepared and simplified in order to reduce the system complexity, and storage and process requirements. It is worth noting that road map preparation module consists of two sub-modules, namely segmentation and layering. Aforementioned modules are explained in the following subsections.

4.3.1.1. Map Preparation Module

In order to find border nodes and overall overview of road map, and also to reduce system overhead and complexity, map preparation modules includes two sub-modules, namely segmentation and layering, which are explained in the following. Using a distributed centralized management system (instead of one centralized system) is another advantage of these sub-modules. Our proposed layered and segmented model which consists of the following four different bottom-up layers is illustrated in Figure 4.2 and explained as follows:

1. *Physical layer*: This layer shows the real road map and nodes corresponding to intersections and junctions with links corresponding to streets and highways. This map can be exported from map databases such as Topologically Integrated Geographic Encoding and Referencing system (TIGER) and OpenStreetMap (OSM). In this layer, the road map is converted to a graph and this graph is given by $G_p = (N_p, L_p)$, where N_p and L_p are the set of nodes and links, respectively.

The segmentation sub-module happens in this layer and divides road map into number of segments with different sizes. Specifying the segments size is based on the number of nodes (i.e. junctions, intersections) in each segment.

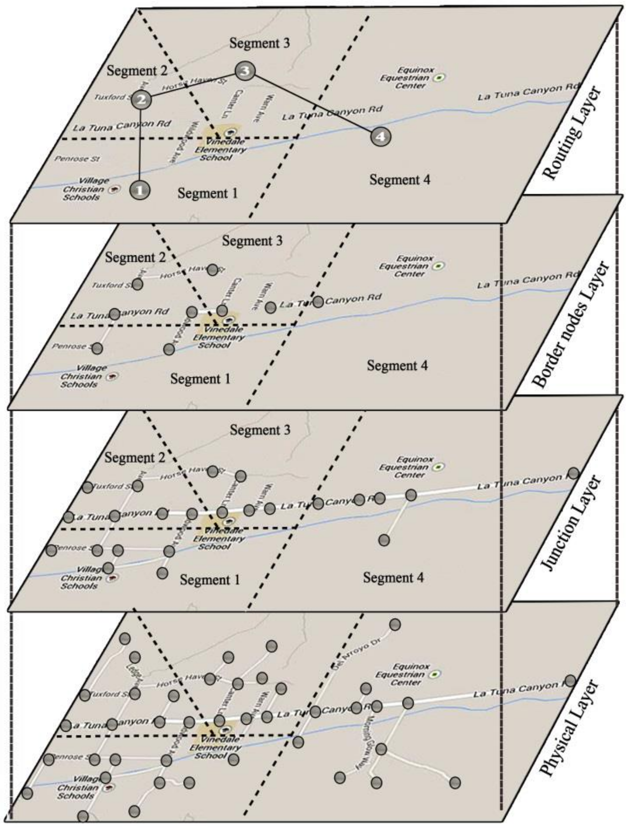

Figure 4.2. Proposed layered and segmented model for AVCAF

In other words, their sizes are assigned in such a way that there is approximately identical number of nodes in each segment. Different sizes of segments are considered in order to maximize the use of resources such as processors and storage devices, and also to balance and reduce the routing overhead in different segments. Moreover, dynamic and quick changes of vehicular environments can be managed using map segmentation and routing is accomplished for each segment individually instead of the whole map. This segmentation is applied in physical layer and each segment is managed by one navigator. Navigators are responsible for creating and updating the routing table for their own segment, which is called the Intra Segment Table ($Intra\_ST_{(i)}$), using an assigned weight to each link in the graph based on the pheromone update rule (Equation 4.8), where i is the segment number (or identifier). $Intra\_ST_{(i)}$ includes m smaller tables where m is the number of nodes in segment i. Each of these smaller tables consists of at least m rows and at most m*n rows as a possible destination, since AVCAF computes up to n alternative paths (i.e. at least one path and at most n paths) between various OD pairs of its own segment and 3 columns that include destination node, next node and route cost. *Intra_ST*$_{(i)}$ and its smaller tables are depicted in Figure 4.3.

Intra_ ST$_{(i)}$

Table 1

Destination	Next Node	Route Cost
1	1	0
2	5	0.9
. . .	. . .	. . .
n	m	0.8

Table 2

Destination	Next Node	Route Cost
1	3	0.5
2	2	0
. . .	. . .	. . .
n	h	0.8

.
.
.
.

Table *n*

Destination	Next Node	Route Cost
1	3	0.5
2	5	0.9
. . .	. . .	. . .
n	n	0

Figure 4.3. An example of *Intra_ST*$_{(i)}$

It is worth noting that smaller tables' index indicate source node (i.e. Table 1 in Figure 4.3 is routing table for node number 1 in segment *i*). Moreover, various values can be considered as route cost such as link length, vehicle traffic condition and selection probability. *Intra_ST*$_{(i)}$ is updated by using our proposed ant-based algorithm in AVCAF which is discussed in the following subsections. Besides, *Table of Segment (ToS)* is created in segmentation sub-module which includes the nodes' name, *ID* and their segment in order to give a complete road map view to all navigator servers. This table is distributed among all servers and helps them to detect the destination nodes' segments in routing procedure. An example of *ToS* is depicted in Figure 4.4.

Node Name	**Node ID**	**Segment ID**
KLCC	1	1
Imbi	2	1
. . .	. . .	. . .
Bukit Bintang	35	5
Wisma R&D	36	6
. . .	. . .	. . .
Raja Chulan	107	7

Figure 4.4. An example of ToS

One of the main requirements of VTRSs, using decentralized processing system, is achievable through the segmentation sub-module. It means that several centralized servers (navigators) are distributed throughout the segmented road map (i.e. one server for each segment), namely distributed centralized servers, to guide the vehicles to their destinations. In this way, road map searching time and the size of routing tables are reduced significantly. This is because each navigator is released from maintaining the whole map information and maintains only small routing tables with local information. Using different strategies (i.e. different types of ants, pheromone update rules, pheromone trail laying rules and probability functions) for different segments is another advantage of road map segmentation in AVCAF.

2. *Junction layer*: Ineffective nodes which do not correspond to a junction or connect two different segments, and their related link(s) are eliminated on the junction layer due to identification of an impossible turning. Junctions are the most critical points in the vehicle routing process since they are the points from which the drivers can be rerouted to a new path.

3. *Border nodes layer*: The nodes on junction layer and their links which connect two different segments in junction layer are retained, otherwise, they are pruned. The remaining nodes (junctions) are called border nodes. The Border Nodes Table ($BNT_{(i)}$), which stores border nodes information, is created for each segment and used for routing to surrounding segments. The *BNT* of each segment is disseminated among all junctions of same segment. Thus, this layers information can be used whenever the source and destination of a vehicle are not in the same segment, but there is a direct link between them. An example of $BNT_{(i)}$ is illustrated in Figure 4.5. $BNT_{(i)}$ has *m* which is the number of border nodes for segment *i* rows and 3 columns that include border nodes' indexes, neighboring segment and route cost. For instance, depicted table in Figure 4.5 indicates that segment *i* has a direct link (road) to segment *j* through its border node with index 20 and with the cost of 0.5.

Segment *i* border nodes indexes	**Neighbouring segment**	**Route Cost**
20	Segment *j*	0.5
37	Segment *d*	0.9
. . .	. . .	. . .
M	Segment *n*	0.35

Figure 4.5. An example of $BNT_{(i)}$

4. *Routing Layer:* The information in this layer is used whenever a vehicle travels over long distances and thus traverses more than one segment to reach its destination. To achieve this goal, a node is assigned to each segment and a link is added between two nodes if there is a link between these two segments' border nodes. An Inter Segment Table (*Inter-ST*) is created based on this new graph (G_R) and is used for distant destination routing (e.g. between segments). $G_R = (N_R, L_R)$, where N_R is the set of nodes assigned to each segment, therefore their numbers are equal to the number of segments, and L_R is the set of links between N_R. The *Inter-ST* is adjacency matrix of

G_R and disseminated among all segments navigators to give an overall view of the map to the distributed navigators. Given G_R with vertices {v1, v2, . . . , vn}, we define its adjacency matrix as follows:

$$\textit{Inter-ST} = \begin{cases} \mathrm{a(i)(i)} = 0, \\ \mathrm{a(i)(j)} = \infty, \text{ if there is no link between i and j,} \\ \mathrm{a(i)(j)} = 1, \text{ if there is link between i and j.} \end{cases}$$

An example of *Inter-ST* is illustrated in Figure 4.6.

Segment index	**1**	**2**	**.**	**.**	**.**	**N**
1	0	1	.	.	.	1
2	1	0	.	.	.	∞
. **.** **.**	. . .	. . .	0 . .	. 0 .	. . 0	. . .
N	∞	1	.	.	.	0

Figure 4.6. An example of *Inter-ST*

Three different cases may occur in the routing process of a vehicle from a node *i* (source) in *X* segment to node *Y* (destination) in *Z* segment:

1. The source and destination nodes are in the same segment;

2. The destination node is within one of the source nodes surrounding areas;

3. Other (Neither case 1 nor 2).

It is worth noting that servers detect the segments of sources and destination nodes through ToS. Therefore, in case 1, the $Intra\_ST_{(x)}$ is used by the navigator for each routing decision. In case 2, the $Intra\_ST_{(x)}$ is used to guide the source node to a border node of the same area. Then the $BNT_{(x)}$ is used to guide the source node to a border node of the destination segment. After that, the $Intra\_ST_{(z)}$ is used to guide the source node to a destination node within the destination area. In case 3, a similar strategy as in case 2 is used, however the *Inter-ST* is used to guide the source node from the previously selected border node to the proper border node of the next segment until it reaches its destination segment. All of these routing tables (i.e. $Intra\_ST_{(i)}$, $BNT_{(i)}$ and *Inter-ST*) are updated using the ant-based algorithm which is discussed in following section.

4.3.1.2. Data Gathering Module

Date gathering module is the second module of initialization phase AVCAF framework. It is one of the main parts of our developed framework since real-time data are collected via this module and having accurate real-time data is a necessity for having a successful and robust VTRS. Ants are the cornerstone of AVCAF framework, especially in data gathering module. We discuss the various types of ant agents used in AVCAF in the following subsections. Vehicle as ANT (VANT) and Packet as ANT (PANT) are the two main types of ant agents which are modeled and used in AVCAF. VANTs are utilized in data gathering module, while, PANTs are further divided into two types, namely Forward ANT (FANT) and Backward ANT (BANT) and used in the second phase of AVCAF (i.e. Optimal path finding phase (section 4.3.2)). Hence, in this subsection, VANTs are discussed in more details.

a) Vehicle as ANT (VANT)

Vehicles are used as real ants in our framework in order to collect accurate real-time information using VANET infrastructures (i.e. V2I communications). Vehicles send basic data such as *ID*, *time* and *direction* as a beacon message to RSUs, which are located at junctions. Using these data, the navigation servers, which are assigned to various segments of road maps in map preparation module, predict the travel speed of each link (PTS_{ij}) in its segment based on historical (deterministic) and real-time (stochastic) speed information. Based on the studies by Chien and Kuchipudi (2003), Kwon and Petty (2005) Nanthawichit, Nakatsuji, and Suzuki (2003), both current traffic conditions and historical data are required for accurate short-term prediction of the time it takes for a driver from an origin to arrive at a destination. In addition, because vehicles' travelling time, speed and density or in other words the cost or probability of each road/street on the road map are not constant over time, for this prediction, we add a vertical time axis and break the time into discrete intervals ($I_0, I_1, \ldots, I_n$) (i.e. 10s), where

I_k = *[start$_k$, end$_k$]* (i.e. discrete time interval (TI) beginning at time instant *start$_k$* and ending at time instant *end$_k$*) in order to consider the dynamic aspect of these data. As a result, our proposed links probability function (Equation 4.4) for FANTs in the next section is a discrete and time-dependent function and a set of time-dependent cost or probability is assigned to edges (roads). A discrete time dynamic network can be represented as a static network using a time-expanded network model (Köhler, Langkau, & Skutella, 2002), which is a useful implicit tool for visualizing, formulating and solving discrete time dynamic shortest path problems. By adopting this model, probability or cast can be satisfactory approximated for each interval.

Historical Travel Speed (*HTS*) and Current Travel Speed (*CTS*) are calculated and assigned to each link *(i, j)* for each TI. If we assume $t = n$ is the current time, *CTS* and *HTS* for link *(i, j)* at time n is calculated using Equations 4.1 and 4.2, respectively, as follows:

$$CTS_{ij}^{n} = \frac{LL_{ij} \times NV_{ij}}{\sum \Delta t_{ID}}\,, \tag{4.1}$$

$$HTS_{ij}^{n} = \frac{\sum_{t=1}^{t=n-1} CTS_{ij}^{t}}{n-1}\,, \tag{4.2}$$

where NV_{ij} is the number of vehicles on link *(i, j)*, and LL_{ij} is the length of link *(i, j)*and has a constant value. Δt_{ID} is the time duration used by specific vehicle (*ID*) to traverse a road between two consecutive junctions. The navigation server of each segment is able to calculate Δt_{ID} and NV_{ij} for its segment's links (roads or streets) as illustrated in Figure 4.7.

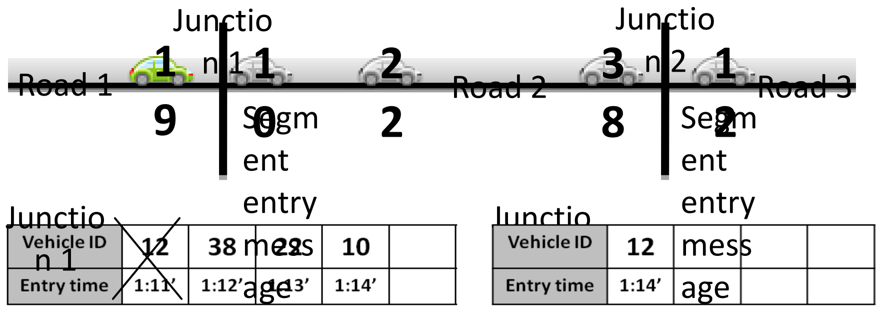

Figure 4.7. Procedure for current average travel speed calculation in AVCAF

NV_{ij} is obtained from the number of vehicle *ID*s in the table of junction i (NV_{ij} is 3 in our depicted example). Vehicles' information is omitted from the junction table upon reaching new a junction and is used by the navigation server to calculate Δt_{ID}. In the example in Figure 4.6, the information of the vehicle whose *ID* of 12 was omitted from junction 1's table and Δt_{12} is calculated as 1:14 – 1:11 min = 3 minutes. The short-term Predicted Travel Speed (PTS) of link *(i, j)* at time *t= n+1* is computed as follows using Equations 4.3:

$$PTS_{ij}^{n+1} = \xi\,(HTS_{ij}^{n}) + \lambda\,(CTS_{ij}^{n}), \qquad (4.3)$$

where ξ and λ weight the effect of historical and current travel speed on the predicted travel speed of roads, and are assigned to 0.4 and 0.6 in this thesis, respectively. It is worth noting that both HTS and CTS are used for PTS calculation in order to consider both recurring and non-recurring congestion, respectively. Delay on a road can be caused by incidents for example by accidents (non-recurring congestion or delay). But there are also regular delays in the rush hours. Regular (recurring) delays exist in historical data. It is important that our framework be sensitive to both of these delays and adapts immediately. That is why Equation 4.3 includes both current and historical speed data. In the non-recurring delay cases that the delays differ completely from the historical data, the adaptation is suboptimal by taking care of historical data. But in most cases the delay will be similar to the historical data so it is good that our framework anticipates the regular delays from historical data even if the current speed does not show any delay. Taking care of only current speed reduces the adaptability of framework. For example, if one driver drives slower than the normal speed it will affect the other vehicles speed as well since its rear vehicles try to change their lane and this lane changing will also affect the vehicles speed on the other lanes. However, since this speed reduction is instant and temporary, its impact on the routing mechanism should be reduced. This can be obtained by considering historical data because real-time data on its own is not adequate for mitigating this issue.

Weighted mean or weighted average concept (Terr, 2004) is utilized in Equation 4.3. Weighted mean is a mean where some values contribute more than others (Finch, 2009). However, the total of the weights is still 1 (i.e. $\xi + \lambda = 1$). Obtained information (i.e. PTS_{ij}) form this module is used by PANTs in Optimal path finding phase in order to find optimal paths between various OD pairs. Optimal path finding phase and its relevant modules are explained in the following section.

4.3.2. Optimal Path Finding

This phase includes two modules, namely map exploration and path selection. After the gathering of real-time information by VANTs in the first phase, PANTs are used to explore and select the shortest least congested paths between source and destination. As mentioned, PANTs are further divided into two types, namely FANT and BANT. FANTs are used in map exploration module, while, BANTs are utilized in path selection module. In addition, we have improved the ant packet header which was proposed in AntNET (M Dorigo & Stützle, 2004) due to the changes in map preparation module (i.e. segmentation and layering). Two new fields are added to represent the source and destinations segments. The new header is used by PANTs during road map exploration and path selection modules. This packet header is illustrated in Figure 4.8.

Packet Type	Address of source node	Segment of source node	Address of destination node	Segment of destination node	Packet length	Packet Sequence Number	Packet Start Time	Packet's Memory	Size of Memory

Figure 4.8. Improved ant packet header

Map exploration and path selection modules are explained in more details in following subsections.

4.3.2.1. Map Exploration

In this module, FANTs explore the road map and construct routes between two specific points (OD pairs). FANTs build a solution by choosing probabilistically the next node to move forward. Different policies can be applied for different segments by using different probability functions considering various user preferences. In this thesis, we proposed a new probability function taking into account vehicles velocity, density, travel time and road length (distance) as user preferences. These metrics are considered in our developed probability function since they have direct effect on vehicle congestion which is our main concern. This probability function (i.e. route cost in *Intra_ST*$_{(i)}$ and *BNT*$_{(i)}$) represented by Equation 4.4 which is used by FANTs for map exploration.

$$P_{ij} = \frac{\alpha(\tau_{ij}) + \beta(\eta_{ij})}{\sum_{\mathrm{h} \in \mathrm{tabu_k}} (\alpha(\tau_{il}) + \beta(\eta_{il}))} \times (\frac{1}{1 + \frac{1}{N_j}}), \tag{4.4}$$

where τ_{ij} (pheromone value) is the learned desirability for an ant in node i to move to node j (next hop) and is computed by BANTs using Equation 4.8. η_{ij} reflects the instantaneous state of the vehicle density and velocity on the link from i to j and computed by VANTs. α and β weight the importance of τ_{ij} and η_{ij}, and are called pheromone power and real-time information power in this paper, respectively. In other words, the impact of gathered data by PANTs and VANTs are tuned by α and β. tab_k is the set of candidate nodes connected to node i that an ant has not visited yet. Finally, N_j represents the number of neighbors for node j. The number of neighbors of next hop, Nj, is considered in Equation 4.4 in order to give higher priority to a node with more neighbors. In this way, the probability of finding new paths increased. FANTs do not deposit any pheromone while moving and do not go to other segments. This allows the use of different types of ants for different segments. The calculation method of τ_{ij} and η_{ij} are explained in the following paragraphs.

η_{ij} is computed using Equation 4.5 by considering vehicle density (D_{ij}) and their average PTS on link (i, j). These two factors are chosen due to their predominant role in vehicle traffic navigation.

$$\eta_{ij} = (1 - D_{ij}) + \frac{PTS_{ij}}{\varphi}, \tag{4.5}$$

where φ is assigned 80 km/h ($\cong$ 22 m/s) which is maximum speed limit for urban and town area in Malaysia (Ong, Mahlia, & Masjuki, 2011). PTS_{ij} is divided by φ to avoid obtaining large values for Equation 4.5. In addition, D_{ij} is calculated using Equation 4.6 as follows:

$$D_{ij} = \frac{NV_{ij}}{Max\_NV_{ij}}, \tag{4.6}$$

where NV_{ij} is the number of vehicles on link (i, j), and Max_NV_{ij} is the maximum capacity of the link (i.e. maximum number of vehicles which can be on the road simultaneously in congested condition) and computed using Equation 4.7.

$$Max\_NV_{ij} = \frac{LL_{ij}}{L_V + \Delta L} \times NL_{ij}, \tag{4.7}$$

where LL_{ij} and NL_{ij} are the length and number of lanes of link (i, j), respectively. ΔL is the average space between two consecutive vehicles. Finally, L_V is the average length of vehicles. ΔL and L_V are considered as 2 m and 5 m in this thesis, respectively (Cheung et al., 2005).

When a FANT reaches its destination, it changes its role and becomes a BANT, instead of dying and copying its memory to BANT (i.e. which happen in most of ant-based algorithms). In this way, the time complexity of the overall system is reduced. The BANT returns the same path as the one traversed by FANT by using its memory but in the reverse direction. The BANT updates the links' pheromone intensity using the pheromone update rule, which is discussed in more detail in path selection module.

4.3.2.2. Path Selection

When all the FANTs have reached their destinations, the pheromone level of each link is updated. This update can either increase or decrease the pheromone trial values. These two phases are called pheromone reinforcement and evaporation in ant-based algorithms, respectively. BANTs use the FANTs memory to return from the destination to the source node. Therefore, they can evaluate the cost of the solutions that they generate and use this evaluation to modulate the amount of pheromone they deposit on the links in return mode. Making pheromone update a function of the generated solution quality can help in directing future ants more strongly toward better solutions. In fact, by letting ants deposit a higher amount of pheromone on the optimal paths, the ants path searching is more quickly biased towards the best solutions. The intensity of pheromone is increased or decreased by using Equation 4.8, which is called the pheromone update rule in our framework.

$$\tau_{ijd}^{new} = (1-\rho)\tau_{ijd}^{old} + \sum_{k=1}^{n} \Delta\tau_{ij}^{k} \tag{4.8}$$

where $\rho \in (0, 1]$ is a constant value, named pheromone evaporation, and n is the number of nodes in the desired segment. The amount of pheromone laid on link i and j by ant k is calculated using Equation 4.9.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{TT_{ij}^{k}} + \frac{1}{D_{ij}^{k}} + \frac{1}{LL_{ij}^{k}} & if\ the\ k^{th}\ ant\ traversed\ link\ (i,j), \\ 0 & otherwise. \end{cases} \tag{4.9}$$

where TT_{ij}^{k},D_{ij}^{k} and LL_{ij}^{k} are the travel time, vehicle density and length of each link of the found route by ant k. As a result, if a link belongs to a found route by an ant, its pheromone value is increased (reinforcement) considering its travel time, vehicle density and length. If it does not belong to a found route, its pheromone value is decreased (evaporates). It means that Equation 4.8 first decreases the pheromone value of all links and then increases it for the links belonging to the found route. Pheromone evaporation improves the exploration factor of the search and encourages the ants to find new routes instead of insisting on the first found route.

Most of the current approaches do not consider accidents in their system due to the complexity of these unpredictable events. In order to consider accidents, we have to deal with two main points: First, the accident: its accurate position, time and status (i.e. how serious it is) should be detected based on collected data via sensors and video cameras or received through reports from other drivers via their smart phones or devices. Second, finding a way to update the routing tables in the shortest time possible with the least delay on the accident condition is a necessity. It is worth noting that some of the current approaches such as the HRS (Tatomir & Rothkrantz, 2006), disable (i.e. ignore) the road where an accident has happened for a while and let the system perform without considering that road in the routing process. The point with the ant-based algorithm is that it handles this situation with some delay due to the stochastic feature of the searching approach and this causes many vehicles to unknowingly join in the congestion before the routing tables are updated. To solve this drawback and to consider accidents implicitly without any concern of the two aforementioned points, TT_{ij}^k, LL_{ij}^k and D_{ij}^k are used simultaneously at Equation 4.9 in AVCAF. By using this new approach, the ants get an additional penalty if they choose the congested roads or the roads with longer travel time which may happen in an accident situation. In this way, roads with less traffic density and travel time are favored even if those roads are somewhat longer.

Another important issue which should be considered in vehicle traffic routing is that, most of the time; all roads are occupied during rush hours making rerouting impossible. Vehicle traffic routing is effective and applicable as long as the full capacity of the roads is not occupied or congested. This issue is considered in AVCAF by finding and utilizing n alternative paths between various OD pairs simultaneously from the early stage of the routing procedure. AVCAF periodically, called re-generation period, Υ, generates pre-defined number of FANTs, N_a, in order to compute and use alternative paths in its routing procedure. FANTs are located on each node of segments as origin points. Then, they start to explore road map using Equation 4.4 and considering the other nodes of their segment as destination points. They find up to *n* alternative paths (i.e. at least one path and at most *n* paths) to the other nodes of their segment. After that, BANTs return to origin points from destination points and update the visited links' pheromone value by using Equation 4.8. $Intra\text{-}ST_{(i)}$ is created and updated by using this information and procedure. It means that different paths with various criteria (e.g. distance, capacity, density, travel time and speed) are used for routing vehicles with the same OD pairs. In this way, road capacities are utilized more efficiently. It is worth noting that these *n* alternative paths are ordered, and got priority based on the probability value calculated by Equation 4.4 which encompasses all the mentioned criteria for each same OD pair. It means that the path with highest probability value gets the highest priority (*n*), while the path with lowest probability value gets the lowest priority (1). Vehicles are routed through these alternative paths in such a way that more number of vehicles is routed to the path with the highest probability value and the fewer number of them are routed to the path with the lowest probability value. Cross-multiplication rule is used in Equations 4.10 and 4.11 in order to identify the portion of routing requests that should be routed through each of the *n* found alternative paths as follows:

$$Z = \frac{100}{n+(n-1)+\cdots+1} \ , \tag{4.10}$$

$$P_n = n \times Z \ , \tag{4.11}$$

where P_n indicates the portion or percentage of the vehicles that should be guided through the path with the priority *n*. Besides, n, $n\text{-}1, \ldots, 1$ are the mentioned priorities assigned to the *n* alternative paths.

This may appear to cause delays for some vehicles because they are routed through longer routes but in fact, it leads to a shorter average travel time for the whole cohort because the congestion is spread to *n* paths instead of only one path. The optimal number of alternative paths (*n*), pre-defined number of FANTs (N_a) and re-generation period (Υ) are obtained through the simulation results.

Ant-based algorithms should be stopped or completed when a predefined condition(s) is reached. A predefined number of iterations, execution time or maximum visited nodes by ants and the pheromone value remaining unchanged for a number of consecutive iterations are some examples of the stopping criteria. However, AVCAF executes for an infinite number of cycles. A cycle completed by reaching a predefined number of iterations where as an ant is dropped by arriving at a predefined maximum number of hops before reaching its destination and is set to $n+1$, where *n* is the number of nodes in a specific segment. It can also be used as an algorithm loop prevention criteria. As a result, besides map exploration and path selection modules, we defined a sub-module, namely stopping procedure in order to consider this issue in AVCAF.

4.3.3. Optimal Path Suggestion

Once the $Intra\_ST_{(i)}$, *ToS*, $BNT_{(i)}$ and *Inter-ST* are created, it is time to route the vehicles based on their destination and obtained information. The abilities of VANET have made it a suitable network for disseminating vehicle routing guidance messages among vehicles. As mentioned in first phase of AVCAF, especially in data gathering module, vehicle-to-infrastructure communications are used for obtaining real-time traffic data from the roads. However, in the third phase, infrastructure-to-vehicle communications are utilized in order to deliver the routing information to vehicles. The procedure of information dissemination for routing services has predominantly been involved with usage of broadcasting (Sun et al., 2000). The reason for choosing this model of propagation is the public interested essence of this information. Hence there have been varieties of approaches proposed with the aim of increasing the performance of information broadcasting from infrastructure (e.g. RSU) to vehicles for specific region of interest. These approaches regardless of their dissemination procedure, have concentrated on increase of transmission rate and coverage area by either usage of multi-hop broadcasting (Busanelli, Ferrari, & Giorgio, 2011; Busanelli, Ferrari, & Panichpapiboon, 2009) or usage of single-hop broadcasting (Sikdar, 2012; L. Yang, Guo, & Wu, 2008). The characteristics of AVCAF as a framework with capability of guiding each vehicle to its destination based on dynamic changes of vehicular environments lead us to modify the procedure of routing information dissemination to single-hop unicast propagation of messages for each vehicle individually. The procedure of optimal path suggestion in AVCAF is as follows:

As mentioned in data gathering module, vehicles periodically send beacon messages to RSUs located on junctions. These messages include the vehicles destination in addition to vehicles' *ID* and *direction.* In this way, the closest RSU can both gather the required data and become aware of the vehicles destinations. The structure of vehicles beacon message is illustrated in Figure 4.9.

Vehicle ID	Message Type	Direction	Destination

Figure 4.9. Vehicle's beacon message structure

In this structure, the "vehicle ID" specifies the ID of vehicle for further communications with RSU, since each vehicle receives routing guidance information individually. "Message Type" specifies the type of message for differentiating the vehicle beacon message using for routing guidance request with other type of beacons that RSU may receive. "Direction" and "Destination" fields specify vehicle's direction and destination which are important for future vehicles topology estimation and vehicle routing. Upon receiving a beacon message, RSU checks the Message Type field. If it is routing guidance request message, RSU adds a row to its database with following information:

vehicle's ID, direction, message received time, destination. Then, it checks vehicle's destination and compares it with *ToS* in order to find the destination node's segment. Based on obtained information, RSU utilizes updated $Intra\_ST_{(i)}$ (i.e. if the vehicle's current segment and destination segment are same), $BNT_{(i)}$ (i.e. if there is a direct link (i.e. road, street) between the vehicle's current segment and destination segment) and *Inter-ST* (if none of the above conditions occurs) to find the proper next junction and its link. Then, it sends this information to vehicle as a message which is depicted in Figure 4.10.

RSU ID	Vehicle ID	Message Type	Next Junction

Figure 4.10. Route guidance message structure

"RSU ID" and "Vehicle ID" indicate the ID of RSU and vehicle, and used for unicast propagation of route guidance message. "Message Type" specifies the type of message for differentiating the routing guidance message from other type of messages that vehicle may receive. "Next Junction" represents the next junction as intermediate or last node that vehicle should pass it in order to reach its destination.

It means that every vehicle receives next junction information before reaching to its current junction via infrastructure-to-vehicle communication (i.e. RSUs located on junctions will send this information). The RSU sends the routing guidance message periodically (using fixed generation rate, 5 message per seconds) up to the time it receives acknowledgement message from the vehicle as a confirmation of receiving the routing guidance message. The reason of using single-hop unicast propagation of messages in AVCAF is that these messages are usable for specific vehicle, since vehicles receive routing information individually. Moreover, it should be delivered to that specific vehicle when it is in close vicinity of a junction and crosses it to another junction. In the following subsection, we give an overview of AVCAF implementation in real world.

4.4. AVCAF in the Real World

To implement AVCAF in the real world, the road map is split into different segments using segmentation sub-module and one server is assigned to each segment. Each navigation server is responsible for a spatially limited area, where it handles routing requests from vehicles within its segment. Vehicle-to-infrastructure communication is a necessity for collecting real-time traffic information. Every crossing vehicle sends some information such as ID, direction and destination to the located RSUs at the junctions. This information is then transferred to the navigation servers which are the cornerstone of AVCAF. Vehicle position is transmitted by GPS-enabled devices (e.g. personal navigation assistant or smart phone) to its nearby RSU. Navigation servers use this information to compute *HTS*, *CTS* and *PTS* for each road within their own segment by using Equations 4.1, 4.2 and 4.3, as explained in Figure 4.7.

Navigation servers regenerate a number of FANTs (N_a) at predefined time intervals, namely re-generation period (Υ) and use them to compute up to n alternative paths (i.e. at least one path and at most n paths) between various OD pairs of its own segment by applying AVCAF. BANTs return to origin points from destination points and update the visited links' pheromone value by using Equations 4.8 and 4.9. $Intra\text{-}ST_{(i)}$ is created and updated by using this information and procedure. *ToS* is used for finding origin and destination nodes' segment. $Intra\text{-}ST_{(i)}$ is used to guide vehicles to their destination if they are within their destination segment, otherwise they are routed to the proper border nodes. Vehicles with a same origin and destination are routed through n calculated alternative paths instead of one path.

The navigation servers communicate via wired networks due to high security and their resistance against interference. They utilize the border nodes and routing layers' (third and fourth layers of our layering model) information to create and update *BNT* and *Inter-ST* in order to find proper border nodes to route the vehicles between two different segments. Using all these data, an OD routing is performed, where each vehicle receives an individual routing guidance based on its current position and destination before each junction in due time by using infrastructure-to-vehicle communication. There is a serious real-time challenge to be solved since each vehicle has its own deadline to receive the routing information based on its speed. AVCAF efficiency is evaluated through simulation in the next sections.

4.5. Conclusion

In this chapter, after reminding the main goal of this thesis, the methodology for solving the problems and achieving the objectives were explained. Quantitative and qualitative methods, which are the most common research methods in computer science field, as well as their characteristics were discussed. By comparing the proposed methodology with these research methodologies, it is concluded that the quantitative method is suitable for our proposed methodology. Moreover, ant-based VTRSs' design challenges are identified and explained in detail, because the main goal of this thesis is designing an ant-based VTRS framework for avoiding vehicle traffic congestion by using VANETs. Based on these challenges and VTRSs' characteristics, we developed a framework, called AVCAF, which includes three phases: 1) Initialization, 2) Optimal path finding, and 3) Optimal path suggestion. In addition, figures and tables are depicted in order to clarify ACVAF procedure and its phases. Every proposed approach should be implemented and then evaluated in order to examine its performance. In the next section, implementation of the proposed framework (i.e. AVCAF) is discussed in detail, whereas its evaluation is discussed in Chapter 6.

CHAPTER 5: IMPLEMENTATION

This chapter is dedicated to the explanation of the implementation details regarding AVCAF framework. Since AVCAF includes VANETs for both real-time data gathering and routing guidance messages dissemination, its implementation, evaluation and testing in real world involve high cost and in most of the cases impossible task if metrics such as testing area, mobility and number of vehicles are taken into account. Besides, most experiments are not repeatable and require high cost and efforts. Therefore, simulation tools and environments are commonly used to evaluate and verify the performance of vehicular networks and VANET based approaches (Mu'azu, Lawal, Haruna, & Rabiu, 2013). Simulation of VANET is totally different form MANETs' simulation due to its unique characteristics and requirements such as dynamic topology, drivers' behavior, multi-path fading, etc. (section 2.1.1). Researchers and developers have built several simulation tools and software for VANETs' evaluation and assessment. It is extremely difficult to choose the proper simulation tool(s) for performance testing without comprehensive and complete analysis of existing tools. Therefore, we give a comprehensive overview of open source simulation tools that allow free access to their source code and explain our selection strategy for simulation tools in section 5.1. After that, we discuss selected tools for AVCAF simulation and evaluation in section 5.2 and its subsections. Section 5.3 discusses about simulation setup including the simulation framework, implementation of AVCAF in NS-2, design of road map, vehicles movements and simulation scenarios in SUMO as well as configuration parameters. Finally the discussion is concluded and sum-up in Section 5.4.

5.1. Introductory to Simulation Tools for VANETs

Existing simulation tools and software for VANETs are discussed in this section due to their importance in implementation and evaluation part of AVCAF. Currently, VANET simulation software can be divided into three categories, namely vehicular mobility generators, network and VANET simulators (Martinez et al., 2011b). Figure 5.1 depicts the taxonomy of VANET simulation software considering the three mentioned categories.

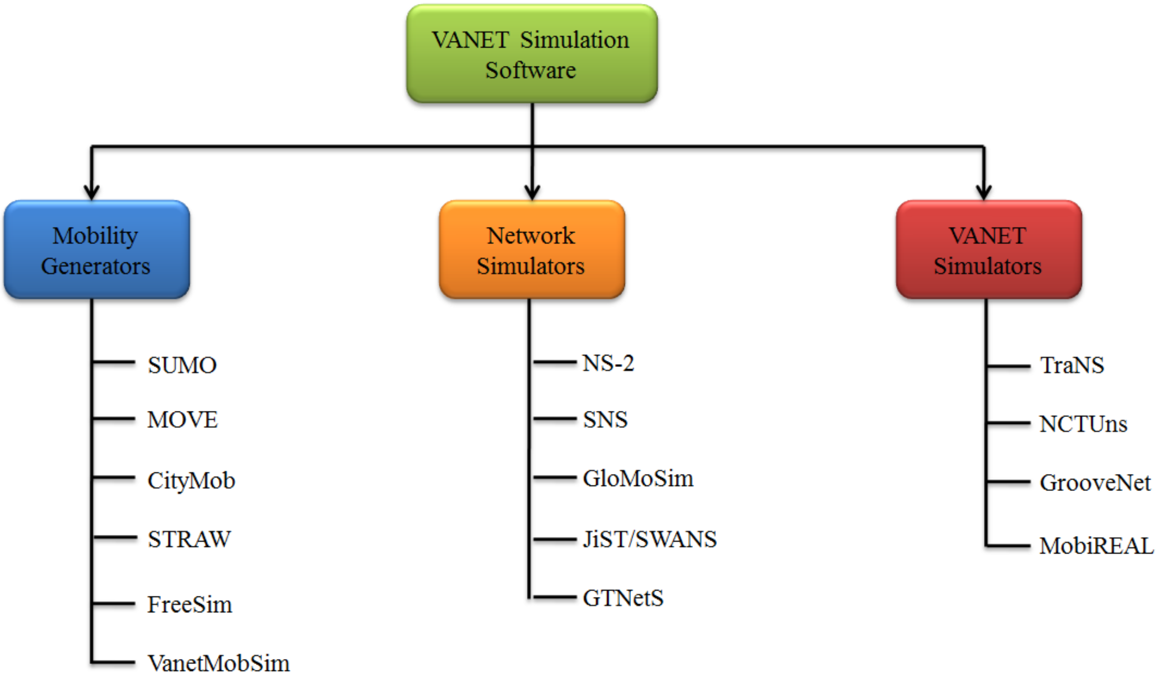

Figure 5.1. Taxonomy of VANET Simulation Software

- Vehicular mobility generators are utilized to enhance the level of reality in vehicular networks simulations. Their generated vehicular mobility traces are used as an input for second category which is network simulator. These inputs consist of road map and scenario parameters (e.g. minimum and maximum speed of vehicles, acceleration and deceleration rates, minimum gap between vehicles, vehicle arrivals and departures rates, and number of road lanes). The output of the trace details the vehicles' positions at every time instant for the whole simulation time and their mobility profiles. The most common examples of vehicular mobility generators are as follows:

1) SUMO (Krajzewicz et al., 2012) is the most widely used open-source, highly portable and time discrete microscopic road traffic simulation package developed for simulating large road maps. SUMO has many features that make it distinguished

form other open-source vehicular mobility generators. Collision-free environment, routing each vehicle individually, different vehicle types, multi-lane streets, right-hand rule routing on junctions and hierarchy of junction types are some of these features. Although many network formats can be imported into SUMO, its generated trace files cannot be directly used by most of the existing network simulators.

2) MOVE (MObility model generator for VEhicular networks) (Karnadi, Mo, & Lan, 2007) is one of the fastest realistic vehicular mobility generators due to its friendly GUI that allows user to create vehicular mobility scenario without writing any script and learning about its internal details. Unlike SUMO, MOVE’s generated trace files can be directly used by existing network simulators such as NS-2 and GloMoSim.
3) CityMob (Martinez, Cano, Calafate, & Manzoni, 2008) allows users to create different mobility models such as simple, Manhattan and downtown model for evaluating purposes. Ability of adding vehicle density similar to real town and inability of importing and utilizing real road maps are the main advantage and disadvantage of CityMob, respectively. Having more than one downtown, multiple lanes for both directions of streets or highways, vehicle queues due to traffic jams are some other features of CityMob vehicular mobility generator.
4) STRAW (STreet RAndom Waypoint) (Choffnes & Bustamante, 2005) is a part of Car-to-Car Cooperation (C3) project (AquaLab, 2002) and creates proper vehicle mobility on US cities maps. Its output trace files are prepared to be used only by JiST/SWANS simulator, which is a serious shortcoming.
5) FreeSim (Miller & Horowitz, 2007) is a fully customizable macroscopic and microscopic free-flow traffic simulator. It allows for multiple freeway systems to be easily represented and loaded into the simulator as a graph data structure with edge weights determined by the current speeds. Designed and developed traffic or graph

algorithms can be performed on all vehicles or for individual vehicle on the road map. Moreover, both user generated data or converted data from real-time traffic data can be used as traffic data in vehicle mobility scenarios.

6) VanetMobSim (Härri, Filali, Bonnet, & Fiore, 2006) is macroscopic and microscopic traffic simulator. At macroscopic level, it can import and use TIGER database or Voronoi tessellation graph as a road map. At microscopic level, it supports mobility models such as Intelligent Driving Model with Intersection Management, Intelligent Driving Model with Lane Changing and an overtaking model, which interacts with Intelligent Driving Model with Intersection Management to manage lane changes and vehicle accelerations and decelerations, providing realistic car-to-car and car-to-infrastructure interactions. Generating output trace files in different formats is one of the main advantages of VanetMobSim. These formats are suitable for most of the existing network simulators such as NS-2, GloMoSim and GTNetS (Georgia Tech Network Simulator).

- Network simulators enable developers and researchers to investigate the networks' behavior under various circumstances. Detailed packet-level simulations of source, destination, reception, route, link, channel and data transmission are accomplished via network simulators. It is worth noting that most of the existing network simulators are developed for MANET simulations. Hence, some modifications and extensions such as vehicular traffic flow model, obstacles and 802.11p standard are needed in order to use these network simulators for VANET simulations. The most used network simulators for VANET simulations via researchers are as follows (Martinez et al., 2011b):

1) NS-2 (McCanne et al., 1997) is a discrete event simulator that includes following features: mobility of node, realistic physical layer along with a radio propagation

model (e.g. free-space, two-ray ground reflection and shadowing model), radio network interfaces and IEEE 802.11 Media Access Control (MAC) protocol via the Distributed Coordination Function. Although the initial versions of NS-2 had some drawbacks in terms of modeling and architecture design of the IEEE 802.11 MAC and PHY modules, Chen et al. (2007) solved these drawbacks by adding some new features to revised architecture of these modules. These features consist of signal to interference plus noise ratio computation, and preamble and physical layer convergence procedure header processing for PHY module, and include IEEE 802.11 carrier sense multiple access with collision avoidance (CSMA/CA) mechanism for MAC module.

2) GloMoSim (Martin, 2001) is developed for simulating both wired and wireless networks. It is scalable, discrete event and layered simulation. Its layers' design is similar to Open Systems Interconnection (OSI) model and standard Application Program Interfaces (API) are utilized between its layers that allow rapid integration of models developed by different developers at different layers. QualNet (Qualnet, 2011) simulator is a trading style of GloMoSim.

3) SNS (Staged Network Simulator) (Walsh & Sirer, 2003) is proposed due to the existing simulators' shortcoming in terms of speed and scalability. The main reason of these drawbacks is that current simulators perform several redundant computations while they can cache the result of expensive computational operations and reuse them as needed. SNS is the staged version of NS-2 and it is 50 times faster than NS-2 when the desired network scenario includes 1500 nodes. This rapid performance allows SNS to simulate large network scenarios. It is worth noting that the existing developed version of SNS is based on NS-2 version 2.1b9a which is not suitable for simulating VANET scenarios.

4) JiST/SWANS (Barr, Haas, & Renesse, 2005) is composed of two modules as its name shows. The first module is JiST which is discrete event and high performance simulation engine. JiST runs on standard Java virtual machine and outperforms existing simulation engines in terms of required time and space for simulating a scenario. It uses virtual machine as a simulation platform via embedding simulation time semantics at the byte code level. As a result, it utilizes Java as its simulations programming language that can be compiled on any virtual machine consist of Java compiler.

   The second module, SWANS, is a scalable wireless networks simulator built on top of the JiST platform. Although SWANS has capabilities similar to GloMoSim and NS-2, it can simulate scenarios that are one or two times larger than what is possible with GloMoSim and NS-2, respectively, using the same amount of consumed time and space for simulation (Rimon Barr, 2004).

5) GTNetS (Riley, 2003) is scalable simulator that is developed for simulating both wired and wireless networks. Like JiST/SWANS, GTNetS is designed to simulate large networks. It is implemented via C++ programming language that allows developers and researchers to extend, develop or examine existing or new protocols through it. It is worth noting that GTNetS consumes lower memory than NS-2. Gathering statistics regarding its own performance and drawing histograms for data sets are the other advantages of GTNetS.

- VANET simulators provide both traffic flow and network simulation. One of the important issues in vehicular networks is the drivers' behavior and response to the VANET applications or messages which can affect the throughput of applications (Sichitiu & Kihl, 2008). VANET simulator allows researchers or developers to change the vehicles' behavior (or drivers' behavior) based on a desired VANET application or message. The most common examples of VANET simulators are as follows:

1) TraNS (Piorkowski et al., 2008) is a VANET simulator that integrates SUMO and NS-2 as a mobility generator and network simulator, respectively, to create realistic VANET simulation scenarios. TraNS is written in C++ and Java, and provides and application centric evaluation framework for VANET. Its latest version, TraNS v1.2, includes new features such as support 802.11p standard, import road maps from TIGER and Shapefile maps, random vehicle routes generation, generate mobility trace file for NS-2, and simulate road events, e.g. accidents. Moreover, it can simulate large networks up to 3000 vehicles and allows for Google Earth visualization of simulation.
2) NCTUns (National Chio Tung University network simulator) (Wang et al., 2003) is an extensible VANET simulator developed for simulating various protocols and algorithms in both wired and wireless networks. A novel kernel re-entering mechanism is used as its main technology. NCTUns can be used as emulator since it supports seamless integration of emulation and simulation. In addition, parallel simulation is achievable via NCTUns on multi core platforms. Its highly integrated and friendly graphical user interface gives many advantages to users and developers such as fast and quick network topology design, variables and modules configuration, nodes movements' specifications and network performance plots. Fedora 9 Linux distribution should be installed on the machines in order to run NCTUns on them which poses a big problem for most of VANET researchers and limits its usage (Martinez et al., 2011b).
3) GrooveNet (Mangharam, Weller, Rajkumar, Mudalige, & Bai, 2006) enables communication between real vehicles and simulated ones by its modular architecture. GrooveNet provides large networks' (i.e. more than 1000 vehicles) simulation along

with new protocols and models evaluation. Three different types of node can be simulated via this VANET simulator: 1) vehicles with multi-hopping data capability over DSRC channels, 2) fixed infrastructure nodes, e.g. RSUs, 3) mobile gateways with V2V and V2I capabilities. Road map scenarios in GrooveNet can be imported from TIGER database, which includes only US maps. This issue is the main disadvantage of GrooveNet and limits its usage by developers and researchers.

4) MobiREAL (Konishi et al., 2005) is developed to simulate realistic nodes' mobility and evaluate MANET applications. It uses C++ programming language and probabilistic rule-based model to create nodes' mobility. This model enables developers to define the relationship between mobile nodes' destination, speed, route and direction, and their position, surrounding area and obtained information from applications. Nodes' movement, their connectivity and packet transmission are visible through MobiREAL Animator that increases the understanding of simulated scenario and obtained results, and makes troubleshooting easier. Collision avoidance among pedestrians and vehicle traffic congestion are considered and implemented in MobiREAL. GTNetS and NETSTREAM (Mori, Kitaoka, & Teramoto, 2006) are used as network simulator and vehicular mobility generator via MobiREAL to simulate VANET scenarios. NETSTREAM is designed and developed by TOYOTA and is a proprietary software. As a result, most of the researchers cannot access and manipulate mobility generator of MobiREAL which limits its wide usage as a VANET simulator.

As it can be seen, several simulation tools and mobility generators are developed for vehicular networks' simulation and evaluation due to their popularity among researcher in recent years. These tools are different in various aspects such as their simulation capabilities, environments, input variables, output trace file formats, available and implemented examples and set of parameters to play with as well as scalability. More information and comparative study of open source simulators for VANETS can be found in the study by Martinez et al. (2011b). In the following section, we explain our strategy for selecting proper simulation tools for AVCAF simulation and evaluation.

When choosing a simulation tools, several questions and aspects should be considered. Installation and learning procedures, input and output variables, ease of usage, available protocols and algorithms, extensibility, size of targeted scenario are some of these questions and aspects for choosing suitable simulation tools. Based on the characteristics of discussed VANET simulators (i.e. TraNS, GrooveNet, NCTUns and MobiREAL) in previous section, GrooveNet, NCTUns and MobiREAL are not suitable for AVCAF development due to the following reasons: 1) Road map scenarios in GrooveNet can be imported from TIGER database, which includes only US maps. 2) Fedora 9 Linux distribution should be installed on the machines in order to run NCTUns on them which poses a big problem for most of VANET researchers and limits its usage. 3) NETSTREAN is used as mobility generator in MobiREAL which is proprietary software and is expensive. As a result, TraNS is selected as VANET simulator for AVCAF development due to following reasons:

- Most of the existing simulator software concern about performance evaluation of packets routing and dissemination protocols, forwarding and MAC protocols related to VANET under realistic but predefined and un-modifiable mobility scenarios. TraNS found a solution for this drawback by proposing Traffic Control Interface (TraCI).
- TraNS uses TraCI (Wegener et al., 2008) to interlink mobility generator and network simulator that enables researchers to control the simulated vehicles behavior during simulation run-time in a real-time manner. As a result, VANET applications can be evaluated in realistic scenarios.

- In addition, NS-2.33 simulator includes a module for testing ant-based approaches (Lima, 2009).
- Last but not least, IEEE 802.11p communication standard (Jiang & Delgrossi, 2008) which is proposed and used for inter-vehicle communication is only implemented in NS-2.33 simulator.

Therefore, it is decided, based on the available unique features, strong focus on vehicular networks, scalability, application-centric mode and having a module to test ant-based algorithm (i.e. AntNet), to use TraNS as VANET simulator in AVCAF implementation and evaluation.

5.2. Simulation Tools for AVCAF Implementation

As mentioned earlier, TraNS integrates SUMO and NS-2 as a mobility generator and network simulator, respectively, to create realistic VANET simulation scenarios. In summary, TraNS, SUMO and NS-2 are selected as VANET simulator, mobility generator and network simulator in order to AVCAF development and evaluation. These three simulation software along with their features and characteristics are discussed in the following subsections.

5.2.1. Network Simulator 2 (NS-2)

NS-2 is developed at the *Information Science Institute* and is sponsored by the *Defense Advanced Research Projects Agency and National Science Foundation*. NS-2 is a discrete event network simulator organized according to the OSI model (Wetteroth, 2001) and was initially intended to simulate wired networks (Hogie, Bouvry, & Guinand, 2006). Afterwards, the 802.11 MAC layer and important wireless routing protocols needed in wireless networks are implemented and added to NS-2 (Schilling, 2005). In general, NS-2 provides a way for users to specify networks protocols and simulating and evaluating their related behaviors.

NS-2 consists of two programming languages, namely C++ and Object Tool command language (OTcl). A huge piece of code and classes written in C++ language composes the core of NS-2. C++ is used for internal mechanism of NS-2 (i.e. backend) due to its quickness and object-oriented specifications. For simulation setup of NS-2 (i.e. frontend) (e.g. objects' configuration and events scheduling) a script language called OTcl is used via NS-2 to make its usage easier. TclCL (Tcl with classes) is used to interlink the C++ and OTcl. A network simulation requires an OTcl script for network configuration, a mobility pattern describing node movement, a traffic pattern describing data traffic, and files describing coordinates for obstacles and pathways. Simulation results which are saved as trace files can be loaded for analyzing by an external application based on the user's preference for each OTcl simulation script: 1) A NAM (Network AniMator) trace file (file.nam) for use with any NS-2 compliant animator tool. 2) A Trace file (file.tr) which has to be parsed to extract helpful information. This procedure is depicted in Figure 5.2.

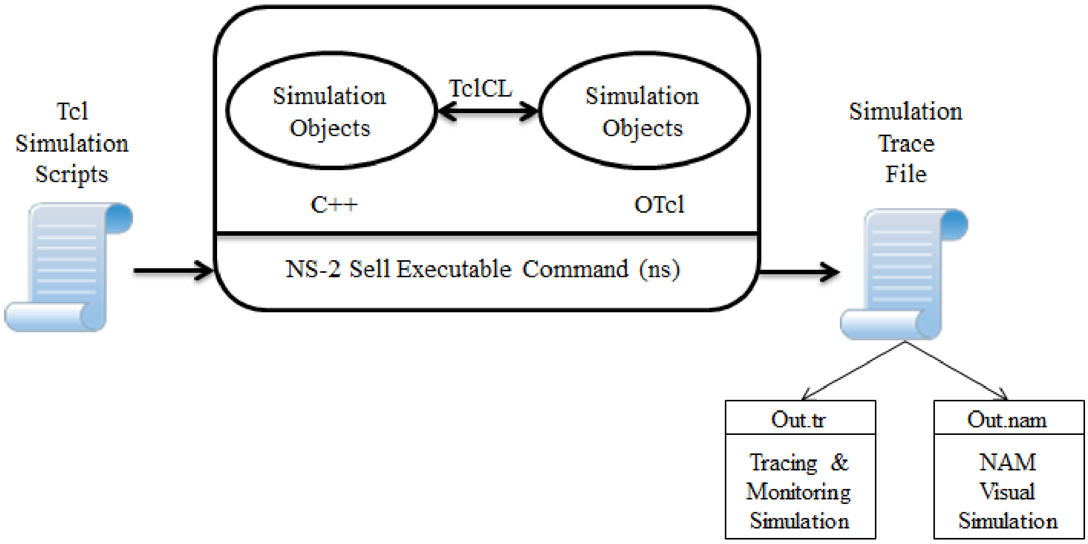

Figure 5.2. Basic Architecture of NS-2

Obtained trace files through the simulation can be processed by user scripts such as Alfred Aho, Peter Weinberger and Brian Kernighan (AWK) programming language or can be shown and interpreted in the form of graph and figures via different tools such as NAM (Estrin et al., 2000) and Xgraph (Harrison, 1989). Up to now, NS-2 is most popular network simulator amongst research communities for simulating and evaluating proposed protocols and algorithms in the area of networking.

As mentioned earlier, NS-2 simulator is a free tool and can be downloaded from The Network Simulator Wiki (2014). Another advantage of NS-2 is that it can be installed on various platforms such as Linux, Windows and Mac systems (Issariyakul & Hossain, 2011). NS-2 can be downloaded and installed in two formats: 1) all-in-one suite and 2) component-wise. In former, all required components along with optional components are embedded in the installation package, while, in the component-wise package, components can be selected and installed by users based on their requirements. The second installation format reduces downloading time and required space for installation. Required and optional components of latest all-in-one installation package are depicted in Figure 5.3.

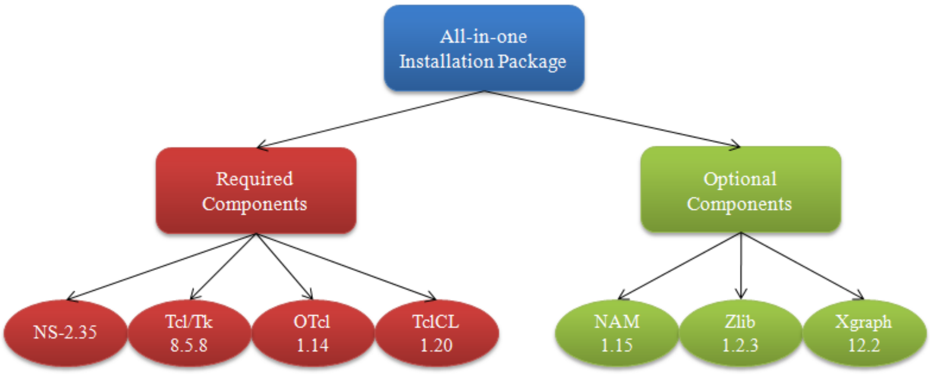

Figure 5.3. Components of the latest version of all-in-one installation package of NS-2

It worth noting that Tcl/Tk is combination of Tcl and graphical user interface Toolkit (Tk). Tk is superior to current methods in developing desktop applications. In addition, Zlib is required library for NAM component.

5.2.1.1. Simulation Steps of NS-2

In order to simulate any protocol or algorithm in NS-2, we need to have an overall view from simulation steps of NS-2. There are three main simulation steps in NS-2, namely simulation design, configuring and running simulation and post simulation processing. These steps are illustrated in Figure 5.4 and discussed in the following paragraphs.

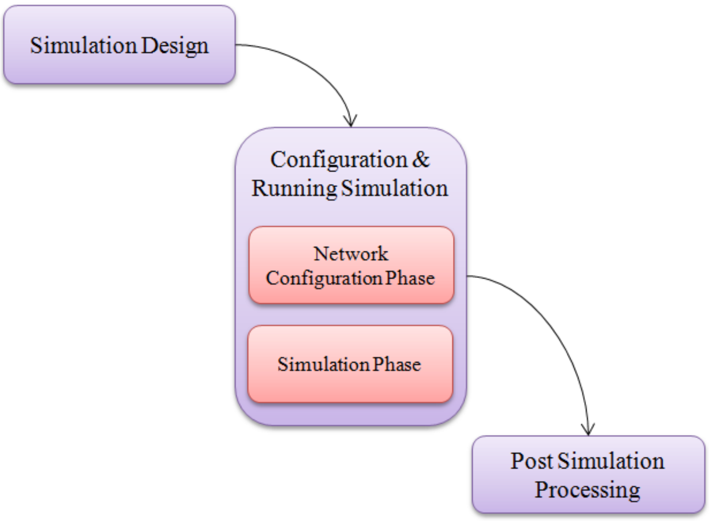

Figure 5.4. Main simulation steps in NS-2

1) *Simulation design*: This is the first step in a network simulation. The simulation users should define the following metrics: the simulation purpose, network configuration, assumptions, the performance measures, and the type of expected results.
2) *Configuring and running simulation*: The above simulation design is implemented in this step. This step is further divided into two phases, namely network configuration and simulation.
   a. *Network configuration phase*: network components such as node, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), and initial events' chain are defined and configured based on designed simulation in the first phase. All the events such as File Transfer Protocol are scheduled to start at predefined times. This phase corresponds to every line in a Tcl simulation script before running the *Simulator* object.
   b. *Simulation phase*: This phase starts the simulation which was configured in the previous phase. This phase corresponds to a single line in a Tcl simulation script. This line invokes the run class of simulator. In other words, simulation phase

maintains the simulation clock, and dispatches and executes all the pre-defined events in chronological order until the last event.

3) *Post simulation processing*: This step includes two tasks, namely debugging and compiling the obtained results via the simulation phase. The first task means the program integrity verification, while, the second one means simulated network performance evaluation. More information about NS-2 simulator and its steps and concepts can be found in the book written by Issariyakul and Hossain (2011).

After successful installation of NS-2.33 and in order to become familiar with ant-based algorithms' simulation and implementation in NS-2, we have implemented AntNet algorithm on NS-2 using manual documentation prepared by Lima (2009). After successful installation and running the AntNet algorithm on NS-2, various setups and modifications have been applied in order to implement and simulate AVCAF in simulator environment. These setups and modifications are discussed in Section 5.3.

5.2.2. Simulation of Urban MObility (SUMO)

SUMO is a fast and time discrete vehicle traffic generator with microscopic features. The major part of development of this software is undertaken by the Institute of Transportation System at the German Aerospace Center. However, every developer or user can contribute in SUMO development because it is open source software and is licensed under GPL. SUMO can be used along with network simulators such as NS-2 or OMNET++ to create realistic simulation of VANET scenarios. SUMO is able to provide a large mobility networks with various features including the simulation of different vehicles, importing the real world maps, providing various traffic characteristics, speed limits, traffic lights, variety of junction layouts and etc.

The microscopic feature of this simulator allows to model variety of vehicles types with different accelerations/decelerations, lengths and maximum speeds. Collision free movements for vehicles, friendly graphical user interface, time discrete vehicle movement and various right of way rules at intersections are the other prominent features of this vehicle mobility generator.

Roads, streets and highways connect to each other to create and form a road map. In other words, we have to create streets and roads as the smallest part of a road map and connect them to each other in for implementing any traffic mobility scenario. In SUMO street consists of two nodes (junctions) and one edge as a street between these two nodes. Moreover, each street or road may have its own properties and rules. Therefore, there are four different file types, namely node, link, link property and lane connection/traffic movement files for designing and implementing any road map in SUMO. The two first files include the information about the junctions (nodes) and the streets (links) between them. Node and link files are saved and named with extensions *.nod.xml* and *.edg.xml*, respectively. Besides, the two other files with extensions *.con.xml* and *.typ.xml* are required in order to specify allowed traffic movements, lane connections at intersections and link types.

First of all, we have to determine the required number of nodes and the links between them. It means that road map should be converted into a graph with nodes and edges and then following files should be created.

- *Node file*: Nodes correspond to junctions on the road map with specific and pre-defined *id* and *coordination*. These coordinates are specified by two-dimensional numeric values for positioning in the desired points as the nodes' latitude.
- *Link file*: The creation of each edge (as a road segment for connecting the junctions) requires election of two junctions which are defined in Node file. Link file includes edges' information such as id, type, direction, start and end nodes (junctions).
- *Link property file*: Additional information regarding each edge or link such as lanes priority based on traffic regulations, number of lanes, and speed limit of lanes can be defined in additional file named Link property file.

- *Traffic movement file*: In order to change the default settings of SUMO such as allowed U-turn or connection between the rightmost lane of each link as well as to specify traffic movements and lane connections an additional file named traffic movement file is required in SUMO.

After preparing four aforementioned files according to traffic scenario and road map, it is required to generate the Network file from them to make the conformity of configured files with SUMO traffic generator format. NETCONVERT command line is used for achieving this objective. Figure 5.5 illustrates the sample of using NETCONVERT command for generating the SUMO network file entitled "*XXX.net.xml*". As it can be seen *.net.xml* extension is used for Network files in SUMO.

```
Netconvert –node-files=XXX.nod.xml –edge-files=XXX.edg.xml –output-file=XXX.net.xml
```

Figure 5.5. An Example of Netconvert Command Line

It is worth noting that SUMO can generate road networks and maps either by using an application called "*netgenerate*" or importing a digital road map via "*netconvert*" as follows:

- *Netgenerate*: This application is a tool for generating different types of maps. The NETGENERATE eliminates the requirement of configuring the Node and assist in constructing the three kinds of abstract networks including the manhattans-alike "grid network", circular "spider network" and "random network". Figure 5.6 illustrates these types of networks.

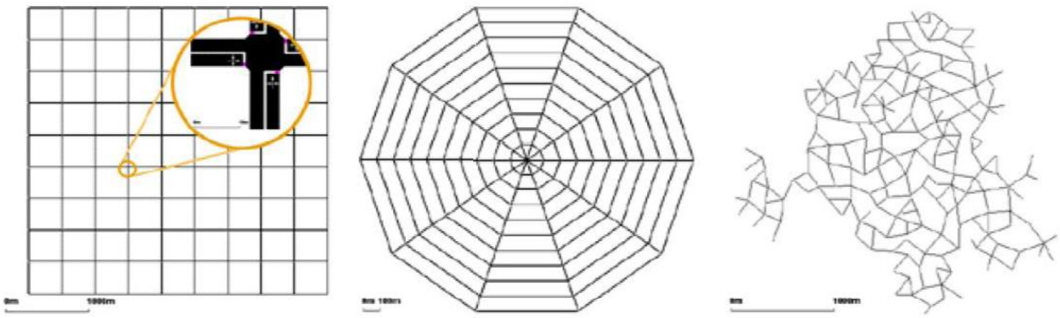

Figure 5.6. The outline of Grid, Spider and Random Networks from Left to Right

- *Netconvert*: As mentioned before, Netconvert is a command line application that enables the users and developers to build various road network topologies. It can be used for generating simple road maps by using node and link files or for generating complex and real road maps by importing them from other traffic simulators such as VISUM (PTV, 2006) and MATSim (Multi-Agent Transport Simulation (MATSim) hompage, 2008) or from open source digital maps such as shapefiles (Stabler, 2006) and OSM (Bennett, 2010). The OSM project as the most cited and used digital maps source using by NETCONVERT.

After creating Network file (i.e. *XXX.net.xml*), Traffic demand file is another required file in order to generate the traffic simulation scenario in SUMO. Traffic demand file includes any information regarding the movement of vehicles beside their types, their quantity, their features and the routes that are required to utilize in the provided network file. Traffic demand file should be stored by *rou.xml* extension.

After obtaining the Network file and defining the Traffic demand file, we need to glue all the prepared files together into a file, called configuration file, which includes both rou.xml and .net.xml. files as well as simulation duration. Configuration file should be stored by *sumo.cfg* extension.

Traffic simulation execution can be achieved by calling the configuration file in two ways, namely SUMO and GUISIM (SUMO-GUI). SUMO application is a pure command line application for efficient batch simulation, while, GUISIM (SUMO-GUI) is the extended application for SUMO which provides the graphical user interface for the simulation. The SUMO-GUI assists the user to observe and monitor the simulation in action. This visual application can be customized in order to show the vehicles speed and waiting time or to follow the traffic behavior of specific vehicle. Polygons, Point Of Interest and image decals are some other graphical features that are exist in GUISIM (SUMO-GUI) to enhance scenarios' visual appearance. It also brings the possibility of interaction with scenarios by changing the prepared traffic signal programs or rerouting the scenarios.

Several output files can be obtained for each simulation run via SUMO. For all vehicles, there is a written range from simulated inductive loops to single vehicle positions in each step as well as other complicated values such as each vehicle's trip information or aggregated measures for all streets or lanes. In addition, noise or pollution emission as well as fuel consumption can be modeled in SUMO which enables users to evaluate the ecological effects of their proposed protocols or applications for vehicular networks. It is worth noting that all output files generated by SUMO are in XML format.

Figure 5.7 represents an overview of simple traffic scenario implementation in SUMO considering the required files. More information about the SUMO mobility generator can be found in the study by Krajzewicz et al. (2012).

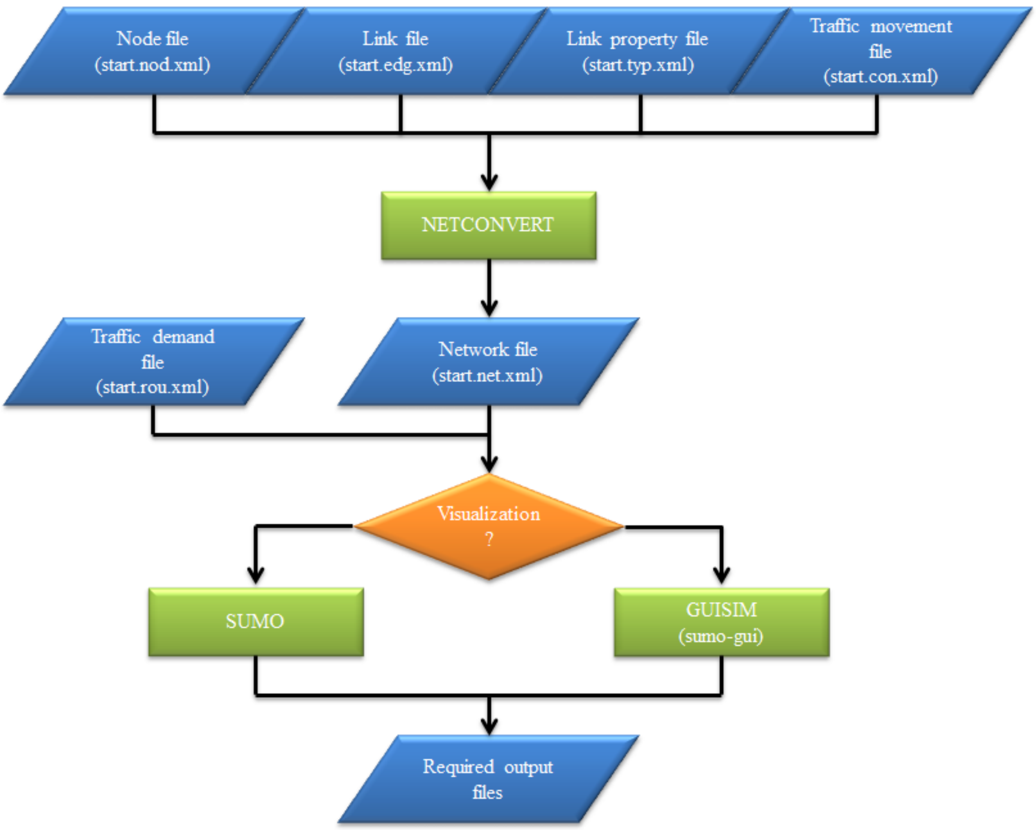

Figure 5.7. General Steps of Traffic Simulation Implementation in SUMO

5.2.3. Traffic and Network Simulation (TraNS) environment

TraNS is a VANET simulator that links SUMO and NS-2 as a mobility generator and network simulator, respectively, to create realistic VANET simulation scenarios. In this way, 1) realistic mobility models can be utilized by NS-2 (i.e. network simulator) and 2) the behavior of vehicles in SUMO (i.e. mobility generator) can be affected and manipulated by NS-2 considering the communication between vehicles. Most of the existing VANET simulators cannot provide the second feature, while the first feature can be obtained by all VANET simulators (Piorkowski et al., 2008). TraNS has two different operation modes, namely network-centric and application-centric. The explanation of each of these modes is as follows:

a) *Network-centric:* Mobility trace files are created and stored on a storage device prior to the network simulator. As a result, this mode can be used for evaluating the protocols that do not effect on vehicles mobility during simulation runtime. User content (e.g. music, file or travel information) exchange or distribution protocol is a suitable example that can be evaluated in this mode.

b) *Application-centric:* This mode enables network simulators to modify the vehicles mobility based on simulated scenario during simulation runtime. Unlike network-centric, the mobility trace files are not created and stored on a storage device prior to the network simulator. This characteristic enables users to create large scale and long-term simulation scenarios without concerning about limited space of storage devices. TraCI is used to interlink mobility generator and network simulator. In other words, both mobility generator and network simulator operate simultaneously and TraCI resides between them and controls their communication. Safety and traffic efficiency applications (e.g. collision avoidance or SmartPark (Piorkowski, Grossglauser, & Papaioannou, 2006)) are proper examples that can be evaluated in this mode. In the following subsection we discuss TraCI in more details since it is one of the main parts of our experimental simulation.

5.2.3.1. Traffic Control Interface (TraCI)

As mentioned earlier, TraCI is an interface to interlink vehicle mobility generator and network simulator in order to control the vehicles mobility and behavior during the simulation runtime. Thus, it enables us to investigate the influence of VANETs applications on vehicular networks. It is worth noting that TraCI is one of the features of latest version of TraNS (TraNS v1.2). TraCI developers believe that any complex mobility pattern, which is a result of an action or a decision taken by a driver, can be fragmented into a sequence of mobility primitives such as ‘change lane’, ‘change speed’, ‘stop’, etc. These mobility primitives can be used to identify the set of atomic mobility commands that used by network simulator to modify and control vehicles mobility pattern.

TraCI is used for creating the bidirectional communication of the SUMO mobility generator and NS-2 network simulator in our simulation experiments. The TraCI provides this interaction over a TCP connection. The TraCI utilizes a client/server based architecture. In which the SUMO and NS-2 software play the role of TraCI-server and TraCI-client, respectively. After establishing a TCP connection, TraCI-client (NS-2) uses data exchange protocol (i.e. commands and responses) to control the simulated vehicles’ movements in TraCI-server (SUMO) based on the designed VANET application. Once a request received, the server (TraCI manager module of SUMO) performs the requested command and sends one or more responses to the client (TraCI manager module of NS-2). Accordingly in each time step the network simulator will influence the traffic simulator and vice versa. During the connection the client periodically sends its simulation time plus one simulation step as a command to the server in order to time synchronization with the server and controls the simulation steps. During the TraCI operation, only the client has the authority to terminate the TCP connection and the server is not eligible for this action. Figure 5.8 illustrates the overview of system architecture interoperation procedure of TraCI.

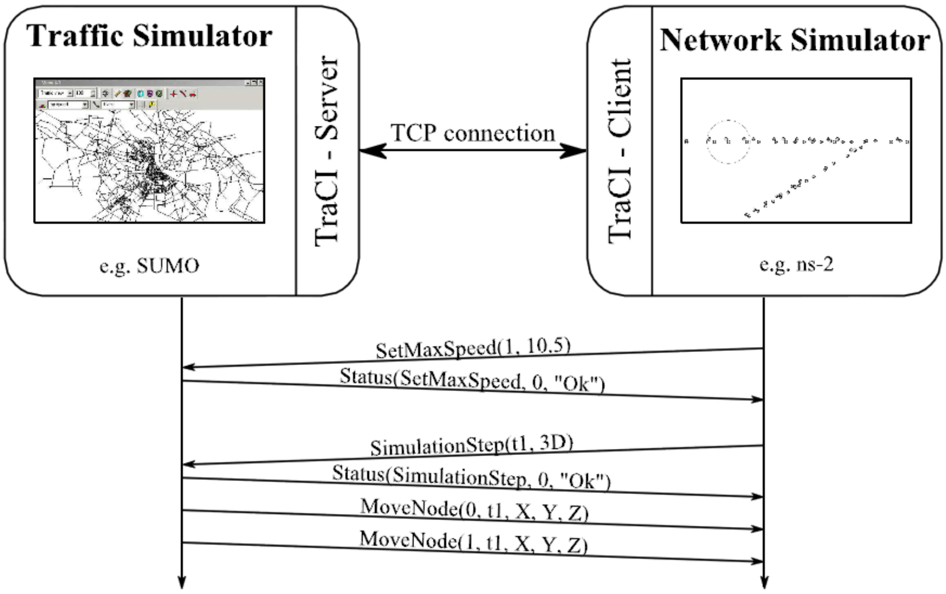

Figure 5.8. Overview of System Architecture Interoperation Procedure of TraCI

The control commands which transfer between the server and client through the TraCI interface can be classified in to three functional types. The first type commands (e.g. simulation setup, status and move node) control the simulation run and are used for permitting the server to execute the simulation up to the next time stamp commands. The second type of commands (e.g. stop node, set maximum speed and change lane) is used for specifying primal atomic mobility behaviours that must apply on the specific vehicle. Finally the third type includes the environmental commands (e.g. scenario, position conversion and driving distance) regarding the road maps, traffic lights, building foot prints, and etc. that are required to provide for the network simulator, considering the context that network simulation has focused. These commands may be the requests for environmental details or instruction for effecting specific vehicle's behavior. More information about the TraCI interface can be found in the next section and the study by Wegener et al. (2008).

5.3. Simulation Setup

This section is devoted to the presentation of the implementation details and simulation setups which are used for developing our approach (i.e. AVCAF framework) by taking advantage of various simulation software including NS-2, SUMO and TraNS as network simulator, mobility generator and VANET simulator, respectively. All of these software were discussed in previous sections. More details about simulation setup, parameters and pseudo-code of the main parts of our approach and simulation procedure that is used for developing the AVCAF are discussed in the following paragraphs. This procedure is depicted in Figure 5.9 to simplify its explanation. AVCAF implementation is explained according to its three main phases, namely initialization, optimal path finding and optimal path suggestion, which were discussed in Section 4.3.

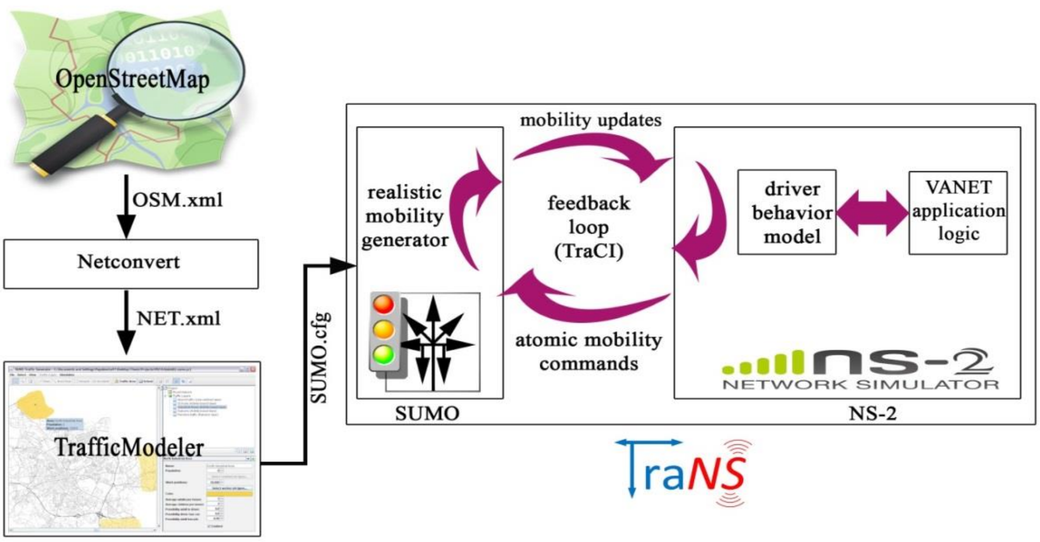

Figure 5.9. Proposed Framework for AVCAF Simulation

5.3.1. Implementation of Initialization Phase

According to AVCAF framework, the initialization phase consists of data gathering and map preparation modules as depicted in Figure 5.10.

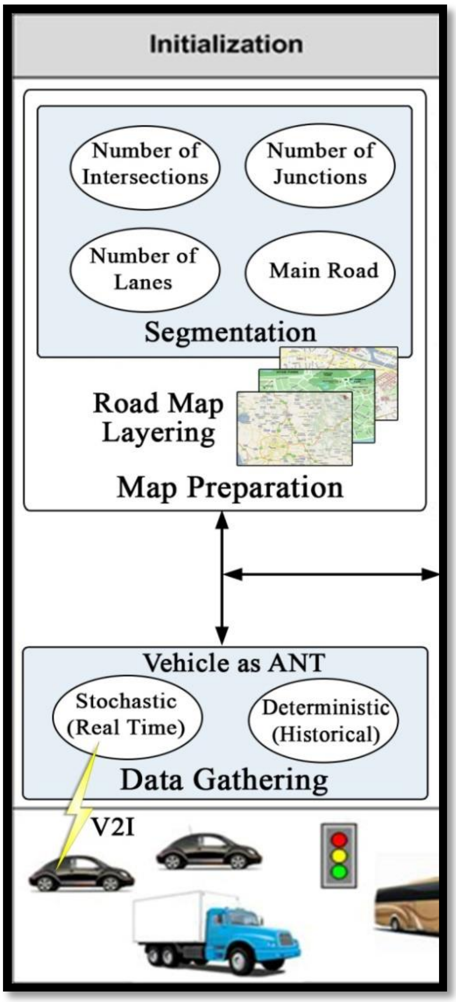

Figure 5.10. Initialization Phase of AVCAF Framework

Regarding map preparation module and providing a suitable and realistic traffic mobility simulation which enables rational evaluation of proposed approach, it is required to design and use a real road map in a manner that possess the prominent feature of urban areas especially the central areas, e.g. downtown. These areas generally comprise orthogonal outline of several adjacent intersections to connect road segments together and are the most common areas that suffer from vehicle congestion problem. For this aim, OSM application is used to export the main structures of real road map in this thesis. OSM is the online editable source of the world map that includes most of road attributes comprise of speed limits, traffic lights, turn restrictions, road types, and etc. OSM provides manual selection of the desired area by using the available option for exporting it as the *.osm* files. A part of the city of Kuala Lumpur, Malaysia map is extracted from OSM in the form of *XML* formatted *.osm* file, namely *map.osm*, and used as the physical road map layer in our simulation which is illustrated in Figure 5.11. Table 5.1 represents the different statistic specifications for this road map.

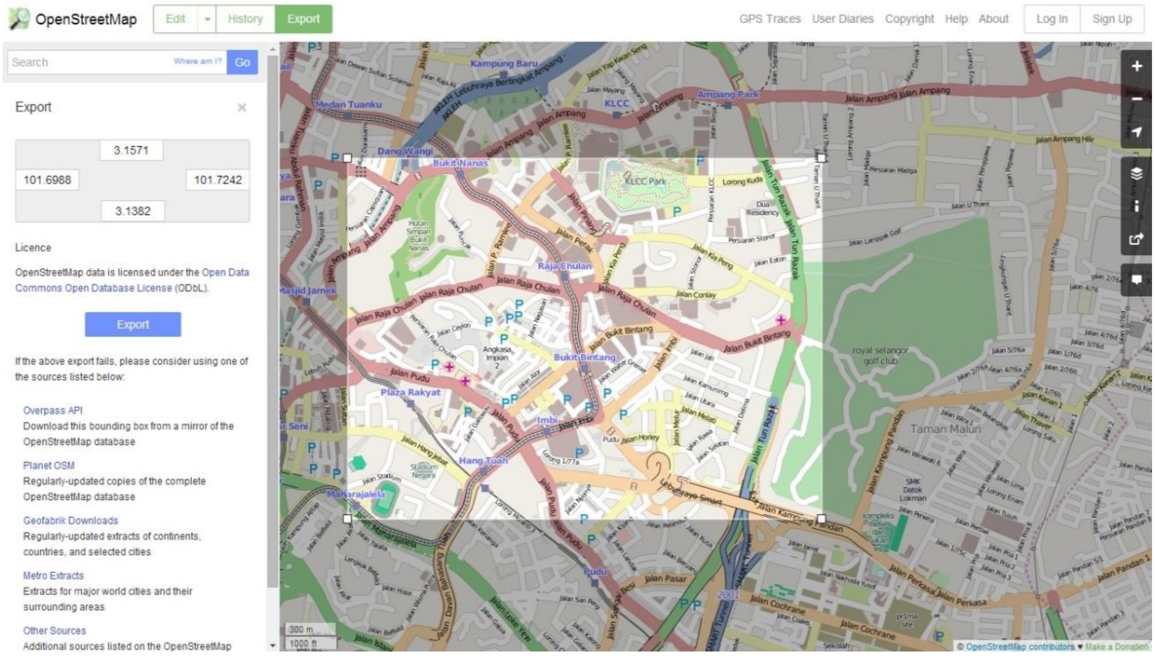

Figure 5.11. The Exported Road Map from OSM Application

Table 5.1. Statistic specifications of road map

Specification	**Value**
Dimension	5 km * 4 km
Map area	20 km^2
Streets/km^2	160.25
Junctions/ km^2	93.3
Avg. street length	225.5 m
Avg. lanes/street	1.9

After obtaining the *map.osm* file, it should be converted into a graph which is suitable format for SUMO. The NETCONVERT tool in SUMO is used to convert the map into a SUMO suitable format (i.e. from *.osm* to *net.xml* file). The NETCONVERT command line that used for converting *map.osm* file into the *map.net.xml* file is illustrated in Figure 5.12.

```
netconvert --osm-files map.osm –output-file map.net.xml
```

Figure 5.12. NETCONVERT Command Line for Obtaining Network File from OSM File

The obtained *map.net.xml* file is imported into SUMO and the result is illustrated in Figure 5.13.

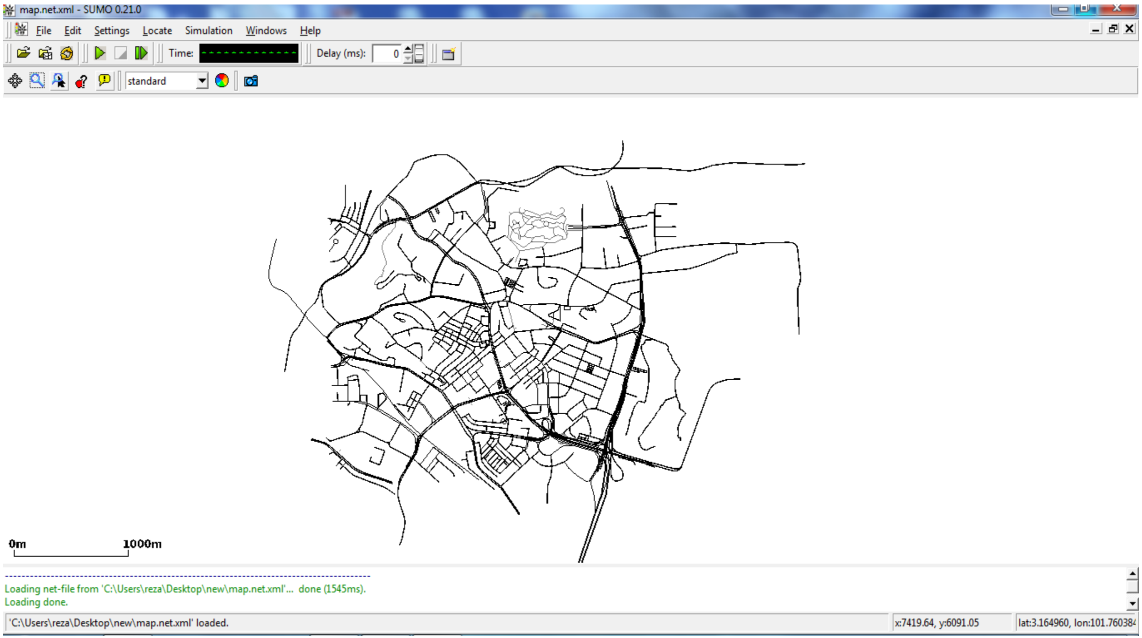

Figure 5.13. The Extracted Road Map in SUMO

In addition to net file, node, edge and connection files are also needed in order to implement the layering and segmentation sub-modules of map preparation. These files are also required for vehicle movement simulation due to following reason. In SUMO, type of each simulated vehicle and its traversed route from origin to destination should be assigned by user. As a result, vehicles movement simulation is very difficult, time consuming and error prone task as the number of vehicles, size and complexity of road map, and variety of designed simulation scenarios such as accident, traffic flow, activity based traffic and weather condition that should be simulated grow. In order to solve this problem, TrafficModeler (Papaleondiou & Dikaiakos, 2009), an open source tool with easy-to-use graphical user interface, is used for quick and high level modeling and generation of vehicles movements in this thesis. Vehicle types, traffic flow elements, emergency conditions are some of the implemented traffic models in TrafficModeler. TrafficModeler uses layering concept for modeling vehicle movements on the road map. Traffic layering concept brings many advantages for us as a user or developer such as simplifying movement model deign, defining various mobility models and assign them to different layers, investigating the impact of various mobility models by enabling or disabling traffic layers, and combining various traffic models designed by different developers. Traffic definition elements are used for defining traffic inside each layer. Each element includes a traffic generation algorithm and a set of attributes that define the way of traffic generation for the desired element based on its attributes such as vehicles' location, departure and arrival time. The current version of TrafficModeler includes three types of traffic layers as follows: 1. User defined traffic: users or developers can use this layer to generate and define their desired traffic model directly via following traffic definition elements: street-to-street, area flow, hotspot and accident. 2. Activity based traffic: users or developers can use this layer to generate and define their desired traffic model indirectly by creating virtual population based on demographical data (Birkin, Turner, & Wu, 2006). Everyday activities of a population and its resulting traffic can be modeled by using this traffic layer via its traffic definition elements, namely traffic area and school. 3. Random generated traffic: users or developers can use this layer to generate and define their desired traffic model indirectly by using some random distribution.

The depicted Netconvert command line in Figure 5.12 is used to obtain *Map.net.xml*. Meanwhile, Netconvert command with –plain-output option is used to obtain the other required files (i.e. *map.nod.xml*, *map.edg.xml* and *map.con.xml*). This command line and a part of these obtained files are illustrated in Figure 5.14 and Appendix A, B, C, respectively.

```
netconvert --osm-files map.osm --plain-output map
```

Figure 5.14. NETCONVERT Command Line for Obtaining TrafficModeler Required Files from OSM File

After importing the road map into TrafficModeler, we generate various vehicle traffic models by using traffic layers, traffic generation algorithms and graphical user interface according to our simulation scenarios (Chapter 6). This step is illustrated in Figure 5.15.

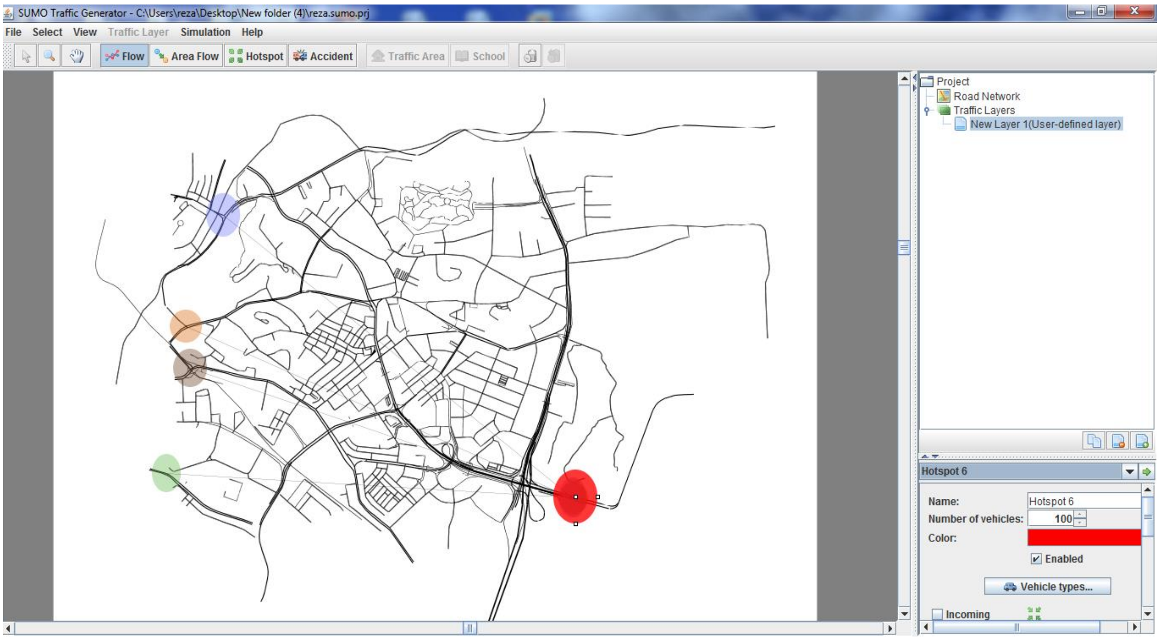

Figure 5.15. Vehicle Movements in TrafficModeler

Generated vehicle traffic patterns are exported and saved in the compatible format with SUMO. In other words, TrafficModeler's output (i.e. *sim.sumo.cfg* and its related files such as *sim.nod.xml*, *sim.edg.xml*, *sim.rou.xml*, *sim.net.xml* and *sim.accidents.xml*) is the input of SUMO.

According to the obtained files (i.e. *map.nod.xml*, *map.edg.xml* and *map.con.xml*), our exported physical road map in Figure 5.11 includes almost 1000 nodes. Hence, it is divided into 20 segments with 50 nodes in each segment based on the segmentation sub-module concept which is discussed in subsection 4.3.1.1. In addition, *ToS*, $BNT_{(i)}$ *and Inter_ST* tables are created based on their described concept in subsection 4.3.1.1 and will be used in last phase of AVCAF framework (i.e. optimal path suggestion phase).

Some other traffic parameters and attributes that are not tunable via TrafficModeler are set and considered manually in vehicle traffic mobility files. These parameters and their assigned value are as follows: speedFactor and speedDev attributes are vehicles expected multiplicator for lane speed limits and deviation of the speedFactor, respectively. 1 and 0.1 are assigned for these attributes, respectively, to result in distribution of chosen ‘maximum speed’ among drivers between 80% and 110% of speed limit. The drivers’ impatient value of 0.7 is considered which cause the partial willingness of drivers to cut into traffic and using gaps in lanes which may not be their right of ways. The driver imperfection (sigma) is set to 0.5 and refers to the inability of a driver to maintain a constant velocity, causing fluctuations in speed that affect the vehicles behind. minGap is the minimum space between two subsequent vehicles and is set to 2 meters. Finally, LaneChangeModel is set to ‘DK2008’ in order to allow vehicles to change their lanes if necessary. These attributes are considered to achieve more realistic traffic behavior.

Regarding data gathering module, the prepared *sim.sumo.cfg* simulation file is imported into SUMO. After running the simulator and assuming the vehicles as ants (VANTs), the stochastic and deterministic data can be obtained via discussed equations 4.1 and 4.2, respectively, in subsection 4.3.1.2.1 and also TraCI commands including EdgeTravelTime and EdgeDensity in subsection 5.3.3. In other words, current density of edge (number of vehicles on specific link) and average travel time of edge will be sent periodically to NS-2 from SUMO via mentioned TraCI commends in order to compute the *CTS* and *HTS* by using equations 4.1 and 4.2, respectively, in subsection 4.3.1.2. Afterwards, *PTS* is calculated by using equation 4.3 and is used for updating routing tables. SUMO provides extensive output options such as travel time, density, mean speed for a given travel time.

5.3.2. Implementation of Optimal Path Finding Phase

The main part of AVCAF framework is implemented in NS-2 simulator. As a result some new files and functions are created in order to obtain this issue. Defining and developing an ant packet, ant agent, routing table and global variables and methods are the most important tasks that are needed for AVCAF implementation. A new ant packet is developed in order to consider segmentation sub-module in addition to source and destination nodes addresses. New ant packet for AVCAF framework is defined in *ant_pkt.h* file which is represented in Appendix D. Map exploration and path selection are two main modules of optimal path finding phase of AVCAF framework as illustrated in Figure 5.16.

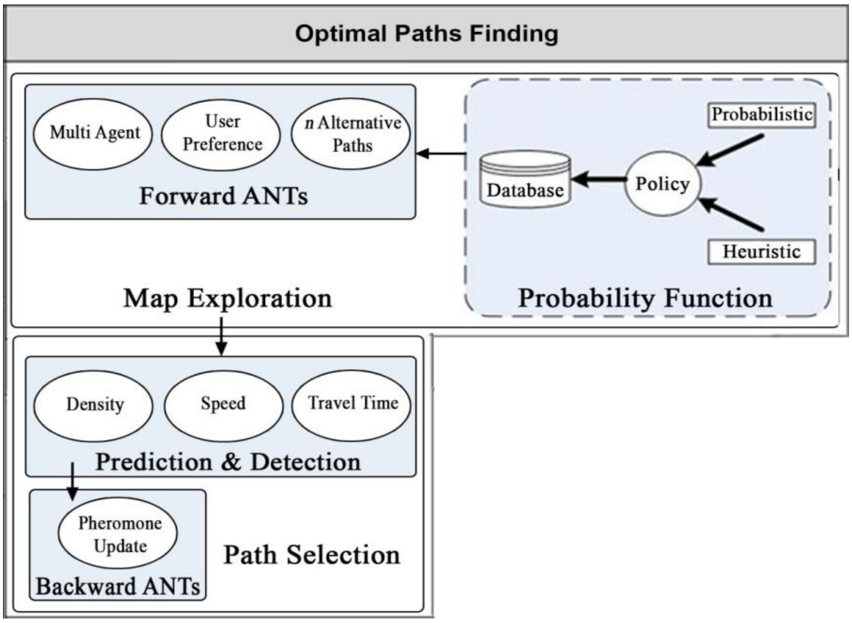

Figure 5.16. Optimal Path Finding Phase of AVCAF Framework

In order to implement map exploration, FANTs are defined in *ant_pkt.h* file. *AVCAF_Intra_ST.h* and *AVCAF_Intra_ST.cc* are definition and implementation files for routing table in AVCAF. Adding an entry in routing table, finding next hop, printing and updating routing table are the most important methods that are developed and embedded in *AVCAF_Intra_ST.cc.* Equations 4.4-4.7 are embedded in finding next hop method of *AVCAF_Intra_ST.cc* file in order to find the next hop address and its probability value taking into account vehicles velocity, density, travel time and road length (distance), number of neighboring nodes as user preferences. Different policies can be applied for different segments by using different probability functions considering various preferences or conditions. For instance, different probability functions can be used for different times of the day because traffic conditions at night-time differ from traffic conditions during the daytime. Hence, higher priority can be assigned to real-time data at night-time comparing its priority at daytime, vice versa; the probability of historical data can be increased during the daytime to increase the gathered data accuracy and efficiency. Although a single probability function (Equation 4.4) is considered in our implementation, two or more probability functions can be utilized in the proposed AVCAF framework based on various conditions. *AVCAF_Intra_ST.h* file is represented in Appendix E. All of these methods take advantage of optimal path finding procedure which its pseudo code is as follows:

```
Procedure OptimalPathsFinding ()
 i - current node, d - destination node, s - source node, N_a - number of ants, γ - re-generation period of ants}
{next_hop - successor node of i, pre_hop - predecessor node of i, P_ij – probability of link (i, j)}
for each node ∈ V do    % V is the set of nodes in the desired segment
   in_parallel    % concurrent activity on each node
      i =s, {nodes – i} = d;
     for each node s do
        if time to generate ant agents at node s then
           for all now and next γ time intervals do
                Create N_a  FANTsd (start_γ)
           end for
        end if
      end for
      for all FANT_sd (P_si) received at node i do
          if i = d {destination reached} then
              Create BANT_ds (P_sd)
              pre_hop ← GetPrev (FANT_sd (P_sd))   % select previous node in the stack
              Move BANT_ds (P_sd) to pre_hop
              Remove FANTsd (Psd)
          else
              next_hop ← GetNext (FANT_sd (P_si))
              Psn ← Psi + GetLinkProbability (i, n, Psi) % compute the link probability using probability
              function (Equation 4.4)
              FANTsd ← Memorize (next_hop, Psn) % add the new information on the ant's memory
              Move FANTsd (Psn) to next_hop
          end if
      end for
      for all BANTds received at node i do
         τdi ← UpdateProbabilities (i, i ← d) % update the pheromone value using pheromone update rule
         (Equation 4.8)
         UpdateInterSegmentTable (i, d, τdi)
         if s ≠ i {source not reached} then
             next_hop ← GetNext (BANTds (Psi)) % select next hop to go to
             Move BANTds (Psn) to next_hop
         else
             Remove BANTds (Psi)
         end if
      end for
    end in_parallel
  end for each
end procedure
```

In addition, ant agent are defined and implemented in AVCAF_ANT.h and AVCAF_ANT.cc files, respectively. These agents are created and attached to each node of road map in order to handle the ant packets and AVCAF implementation. AVCAF_ANT.cc contains methods to send, receive and process FANTs and BANTs based on obtained information from AVCAF, to build memory of FANTs, to add neighbor nodes, to print neighbors of a node and to update the traffic model. AVCAF_ANT.h file is represented in Appendix F for more clarity.

Regarding implementation of path selection module of optimal path finding phase, BANTs are created and forwarded to the next hop based on the FANTs' memory. These two methods (i.e. creating BANT packet and Forwarding BANT to next hop) are embedded in AVCAF_ANT.cc file. BANTs use the FANTs memory to return from the destination to the source node. Therefore, they can evaluate the cost of the solutions that they generate and use this evaluation to modulate the amount of pheromone they deposit on the links in return mode. Pheromone update rule (Equation 4.8) is utilized for this purpose. Link's pheromone value is updated considering its length, vehicles' density and vehicles' travel time via Equation 4.9.

Global parameters such as α, β, ρ, λ and Υ, and methods such as finding number of neighbors of a node as well as number of vehicles on a road between two nodes are defined and implemented in AVCAF_common.h and AVCAF_common.cc files. These files are represented in Appendix G and H, respectively. It is worth noting that in order to find the best value for the various parameters of AVCAF, their impact on AVCAF were examined individually through simulation and the obtained results are explained in Chapter 6.

After creating and implementing required files and methods of AVCAF for NS-2, we modified NS-2 and installed our codes to simulate and evaluate AVCAF. Some of these modifications are as follows: adding ant packet type and name to constructor of class p_info() in packet.h file, adding new function for trace format of AVCAF in cmu-trace.h file, including ant packet header file, defining trace format of AVCAF and adding case for ant packet to CMUT::format() function in cmu-trace.cc file, adding ant packet reception ability by editing PriQueue::recv() function in priqueue.cc file, editing ns-packet.tcl file to add AVCAF in it, adding some default values such as number of nodes, simulation duration, pheromone evaporation rate (ρ) and regeneration period of ants (Υ) in ns-default.tcl file, Editing create-wireless-node method and adding create-AVCAF-agent method in file ns-lib.tcl in order to create an instance of AVCAF agent for a node and editing Makefile to add object files to OBJ_CC variable. After doing these steps, the AVCAF ant packets and agent can be used to run AVCAF framework and obtain $Intra\_ST_{(i)}$ tables via tcl script for vehicles routing. An example of $Intra\_ST_{(i)}$ table is illustrated in Figure 5.17. As it can be seen, Intra_ST table is related to segment 1 and it includes a routing table for each node of its segment. For instance, in Figure 5.17, routing table for node 7 is illustrated considering destination and next hop nodes as well as their related probability values (i.e. pvalue).

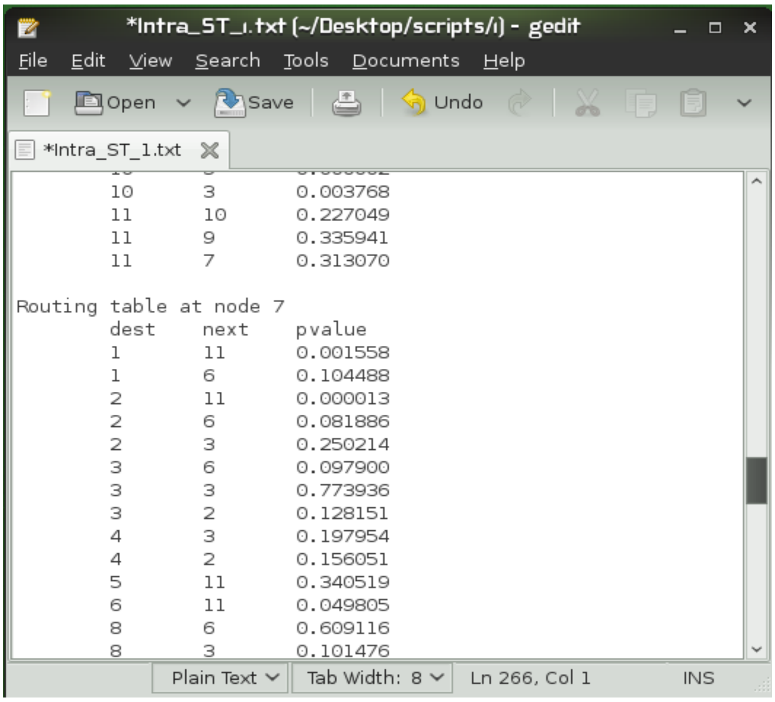

Figure 5.17. An example of Intra_ST table

It is worth noting that in urban scenarios, and at the frequency of 5.9 GHz (i.e. the frequency band adopted by the 802.11p standard), radio signals are highly directional and will experience a very low depth of penetration (Bohm, Lidstrom, Jonsson, & Larsson, 2010; Sommer, Eckhoff, German, & Dressler, 2011). Hence, in most cases, buildings will absorb radio waves at this frequency, making most communications only possible when vehicles and RSUs are in line-of-sight. In order to accurately simulate how radio signals propagate in urban scenarios, we must consider the effect of signal attenuation due to distance, along with the effect of obstacles blocking signal propagation. Therefore, to better simulate wireless signal propagation, both attenuation and visibility schemes should be taken into account (Martinez et al., 2013). The ns-2 simulator, in version 2.33, offers some schemes to account for wireless signal attenuation, but in the case of visibility, none of them support obstacle modeling within the network. Therefore, simulating AVCAF in current visibility scheme of ns-2 simulator leads to an obstacle-free environment simulation which is far from a real environment. We considered the visibility model along with the attenuation model (e.g. Rayleigh, Two-ray Ground and Nakagami model) for simulating obstacles in our simulation environment. For this purpose, we utilized Topology-based Visibility model proposed by Martinez et al. (2013). This model considers road dimension and geometry in addition to line-of-sight in its modeling procedure. In the case of the attenuation model, a probability density function was used to determine the probability of a packet being successfully received at any given distance. With regard to the visibility model, more complex and realistic street layouts such as roundabouts, angled roads, and merged-and-split roads were considered in the Topology-based Visibility model. Each street contour can either be simulated as an empty area or a building wall using this model. At the MAC and physical layers, the IEEE802.11p standard was used for the wireless configuration. We utilized the street broadcast reduction scheme (Martinez, Toh, Cano, Calafate, & Manzoni, 2011a) to alleviate the broadcast storm problem in our simulations. All results represent an average of over 5 executions with different scenarios (maximum error rate of 10% with a degree of confidence of 90%) and the maximum transmission range is set to 400m.

According to the aforementioned discussion regarding setting up and configuring the SUMO and NS-2 simulation tools, the Table 5.2 presents brief description about important parameters values used in both simulators.

Table 5.2. Configuration parameters of SUMO and NS-2

Parameter	**Value**	**Parameter**	**Value**
Road map dimension	4 km × 3 km	Simulation time	450 seconds
Road map area	12 km^2	Size of packets	500 bytes
Streets/ km^2	240.25	Vehicle speed	0 – 30 m/s
Junctions/ km^2	150.3	Vehicle density	100 to 1000
Avg. street length	205.5 m	MAC/PHY	IEEE 802.11p
Avg. lanes/street	1.9	Max. transmission range	400 m
Portion of vehicles for each type	15% freight vehicle	Speed distribution	speedFactor 1
	85% passenger vehicle		speedDev 0.1
Vehicle lengths	5m, 10m	minGap	2m
Driver impatient value	0.7	Gradient	0
Intersection traffic rule	Permissive right turn	Transmission power	13 dB_m
Channel bandwidth	10 MHz	Channel frequency	5.9 GHz
Sigma (driver imperfection)	0.5	LaneChangeModel	DK2008

5.3.3. Implementation of Optimal Path Suggestion Phase

Since the main objective of AVCAF is routing the vehicles through the least congested shortest paths, vehicles receive rerouting information during their travel from origin to destination. As a result, it is needed to modify the vehicles movements according to the updated $Intra\_ST_{(i)}$, $Inter\_ST$, $BNT_{(i)}$ *and ToS* routing tables in the application part which is embedded in NS-2. It means that our developed approach, AVCAF, in NS-2 may decide to reroute some vehicles due to considering recurring or non-recurring congestion on a specific road based on updated routing tables. Therefore, the optimal path should be sent and suggested to the driver on desired road. Optimal path suggestion phase of AVCAF framework is depicted in Figure 5.18.

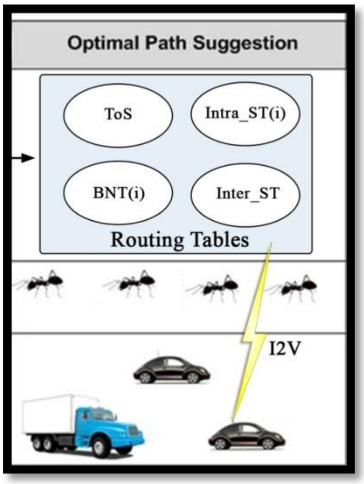

Figure 5.18. Optimal Path Suggestion Phase of AVCAF Framework

In real environment V2I communication can be used for this purpose. However, in simulation environment, the network simulator (i.e. NS-2) instructs the vehicle mobility generator (i.e. SUMO) to change the mobility attributes of some specific vehicles, while the other vehicles attributes remain unchanged and are controlled by vehicle mobility generator only. As mentioned before, we interlink the vehicle mobility generator and the network simulator via TraCI which is a part of application centric mode of TraNS (section 5.2.3). As depicted in Figure 5.9 and discussed in section 5.2.3.1, TraCI provides an active feedback loop over a TCP connection between two simulators (i.e. SUMO and NS-2) that allows modifying the mobility attributes of a specific vehicle or a group of vehicles by exchanging a set of commands or responses according to the updated information from VANET application or algorithm (i.e. ant-based algorithm in optimal paths finding phase of AVCAF). TraCI messages are used to exchange these commands and responses between SUMO and NS-2. TraCI message format is illustrated in Figure 5.19. It composed of a small header which contains overall message length and a variable number of commands that each of them contains commands length, identifier and content.

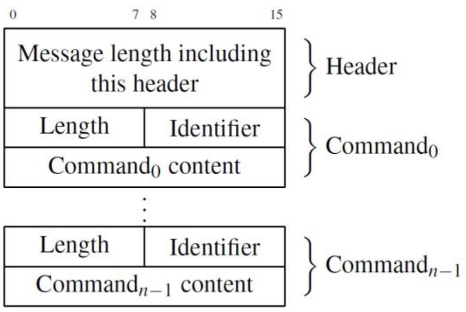

Figure 5.19. TraCI Message Format

TraCI commands that are designed and used in current study for syncing the SUMO and NS-2 are as follows:

- SimulationStep: It is called periodically by NS-2 to retrieve node (vehicle) positions and to make sure the simulation times between NS-2 and SUMO are synchronized. SimulationStep command includes 2 fields, namely TargetTime and PositionType. The first field is used to allow SUMO to perform the simulation until the next time period. It means that TargetTime is set to current time of NS-2 plus one simulation step. The latter field indicates the requested type for node position that can be either 3D or road map position. Status and MoveVehicle are sent by SUMO as a response to SimulationStep command.
- Status: It is sent as a response or acknowledgement to every request command and contains result flag and description fields. The result flag indicates the success or failure of the requested command and its explanation is embedded in the description field.
- MoveVehicle: It contains movement information including vehicle ID, position and TargetTime for one vehicle and is used to transfer the vehicles mobility into NS-2. This information is converted into linear movements by NS-2 in order to make sure that each specific vehicle reaches its desired position at TargetTime.
- EdgeTravelTime: Current travel time of edges (links) will be sent periodically to NS-2 via SUMO in order to compute the Predicted Travel Speed (*PTS*).

- EdgeDensity: Current density of edges (number of vehicles on specific link) will be sent periodically to NS-2 via SUMO in order to compute the Predicted Travel Speed (*PTS*). SUMO provides extensive output options such as travel time, density, mean speed for a given travel time.
- SetMaximumSpeed: It has 2 fields, namely vehicle ID and MaxSpeed and limits vehicle's speed or removes such a limit. SUMO is responsible to control the vehicle's speed limits based on its mobility model.
- SetMinimumSpeed: This command is similar to the previous command (i.e. SetMaximumSpeed). However, it is used for assigning minimum speed to vehicles instead of maximum speed.
- StopVehicle: It is used to stop a vehicle at a certain position for a specific time period. SUMO is responsible for following its mobility models and stops the vehicle at the desired position. StopVehicle command has 3 fields, namely Vehicle ID, StopPositionArea and WaitTime. Since most simulators are time discrete, it is not feasible to stop the vehicle on an exact position. As a result, stoppage area is defined by StopPositionArea around the stop position. The vehicle waits for a period of time (WaitTime) by setting its maximum speed to zero.
- ChangeRoute: This command allows a vehicle identified by Vehicle ID to react to certain traffic conditions by adapting its route. Therefore, when a probability of a road identified by Road ID is changed based on the traffic condition, NS-2 notifies SUMO using this command and a new route is calculated before the simulation continues.
- ChangeTarget: It can be used to change the destination of a specific vehicle (Vehicle ID). The new destination can be a junction, intersection or road. New route to the new destination is calculated before the simulation continues.

- Scenario: This command is used for getting scenario parameters such as map size, and number of nodes, especially at the beginning of the simulation.
- DrivingDistance: This command calculates the traversed distance by an identified vehicle from its origin to destination.
- DrivingTime: This command computes the elapsed time for a vehicle to reach its destination from its origin.

By using these TraCI commands the optimal paths are suggested to drivers based on their destinations and vehicles' movements are modified and changed are changed in SUMO in order to implement this phase.

5.4. Conclusion

This chapter discussed about simulation procedure and implementation of proposed approach, AVCAF, in simulation environment. In order to select appropriate simulation tools for implementation of our approach, a comprehensive overview of open source simulation tools was conducted in this chapter. Based on this information and discussed reasons, SUMO and NS-2 are selected as a mobility generator and network simulator, respectively, to implement our proposed approach and create realistic VANET simulation scenarios. In addition, TraCI is used to interlink SUMO and NS-2 in order to control the vehicles mobility and behavior during the simulation runtime. Accordingly the important modules, components and parameters which are required in designing AVCAF framework were created in NS-2. Afterwards, the implementation and configuration of various parts of SUMO simulator, including the design of real road map by using the OSM and NETCONVERT application, generation of vehicle movements and simulation scenarios by using TrafficModeler software were elaborated. Then important configurations of environmental characteristics, connection manager, MAC and physical layers were discussed. Finally, the overview of all discussed configuration parameters' values for both mobility generator and network simulator were summarized in a table for simplicity.

CHAPTER 6: RESULTS AND DISCUSSION

This chapter goes through explanation of the obtained results from running of implemented simulation scenarios in order to evaluate the effectiveness of our proposed approach. As it was mentioned previously the main aim of this study is to propose an effective framework to avoid vehicle traffic congestion by considering both recurring and non-recurring congestion conditions. Accordingly various scenarios along with different evaluation metrics are defined and designed to ensure that our main aim is achieved by the proposed approach. Moreover, some benchmark approaches are selected to be compared with our proposed approach. As mention in chapter 4, AVCAF includes various parameters such as α, β, ζ, λ and ρ in its routing procedure and its performance is highly dependent to the value of these parameters. Hence, at first step, the proper value for these parameters is found by examining various values for them through the simulation which is discussed in Section 6.1. Benchmark algorithms and approaches that are used for performance evaluation purpose along with their selection reason are discussed in Section 6.2. Simulation scenarios and evaluation metrics are represented in Section 6.3. Section 6.4 includes the obtained results and their discussions for evaluated approaches considering various designed scenarios and evaluation metrics. Finally, Section 6.5 concludes the section by summarizing the main points and findings.

6.1. Simulation results for AVCAF parameters' value

To find the best value for the various parameters of AVCAF, their impact on this framework were examined individually through simulation and explained as follows. All of the results represent an average of over 5 executions with different scenarios (maximum error rate of 10% with a degree of confidence of 90%). Due to the high number of AVCAF parameters and consequently high number of iterations, a simple road map is used in this section. The designed road map is illustrated in Figure 6.1 which includes 12 nodes (junctions) including 2 border nodes, 15 links (roads), and 2 segments. It is worth noting that the length of each link is assigned to 500 m and 3 vehicles are located at each point with random destinations (i.e. number of vehicles = 12 × 3 = 36). The average travel time is utilized as a measurement criterion in this section.

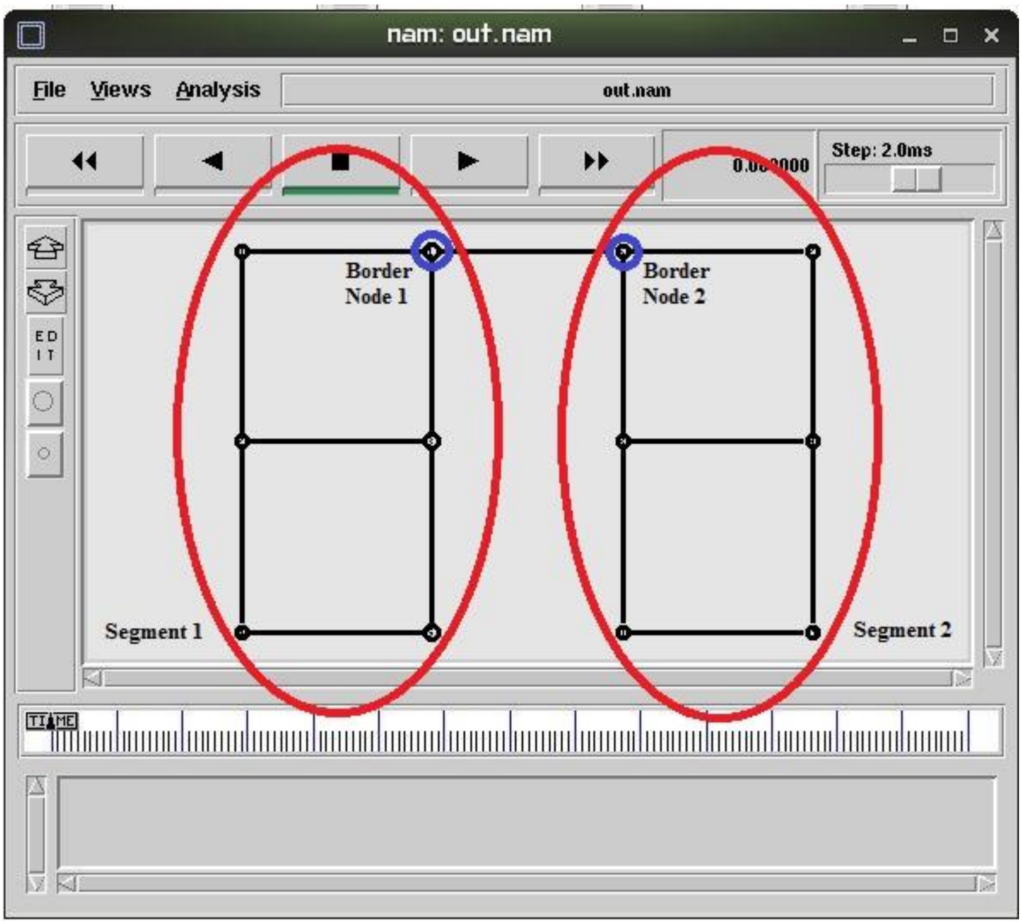

Figure 6.1. Map used for finding AVCAF parameters' values

1) *HTS information power (ξ):* This parameter specifies the effect of HTS information on the short-term prediction travel speed of roads (Equation 4.3 in chapter 4). The historical data influence has increased by raising the value of ξ, while, decreasing the value of ξ will reduce the effect of historical data on the path selection procedure.

2) *CTS information power (λ):* the function of this parameter is very similar to ξ but the difference is that it controls the CTS information impact on the path selection procedure (Equation 4.3 in chapter 4). It is worth noting that this information is gathered by VANTs.

There should be a proper trade-off between ξ and λ (i.e., $\xi + \lambda = 1$, called weighted mean (Terr, 2004)). In our simulation environment, the best condition occurs when ξ =

0.4 and λ = 0.6 for AVCAF evaluation. Figure 6.2 illustrates the average travel time of the found paths by AVCAF as a function of the HTS and the CTS information power, while considering other parameters as follows: α = 0.5, β = 0.5, ρ = 0.5, N_a = 15, n = 3, γ = 5 TIs (50s).

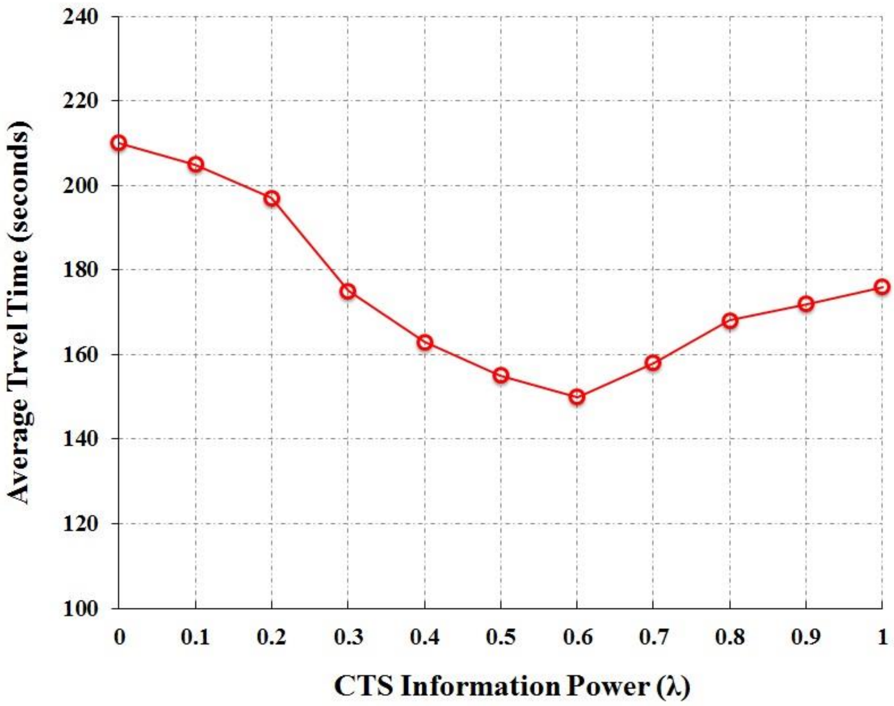

Figure 6.2. Average travel time for AVCAF as a function of CTS information power (α = 0.5, β = 0.5, ρ = 0.5, N_a = 15, n = 3, γ = 5 TIs (50s))

The average travel time converges towards two different values at the beginning (λ from 0 to 0.2) and at the end (λ from 0.8 to 1) of this diagram. This is because at the beginning, path finding is more based on HTS information ($0.8 \leq \xi \leq 1$) whereas at the end; it is more based on CTS information ($0.8 \leq \lambda \leq 1$). Our obtained results, assigning higher value to λ compared to ξ (i.e., λ = 0.6, ξ = 0.4), can be supported by the following reasons: 1) considering non- recurring congestion condition (i.e. accident, working zones, and weather conditions) in vehicle routing is one of our main concerns. Based on the results obtained by Rakha and Van Aerde (1995), the traffic conditions vary considerably from one day to the next day due to non-recurring congestion conditions. Consequently, the historical data (i.e., HTS) will be insufficient for commuters to find the optimum routes through the network, and the provision of current traffic information could provide major benefits, and 2) theoretically and based on the obtained results by Karbassi and Barth (2003), the prediction gets more and more closer to the true value with increasing the number of real-time observations.

3) *Pheromone power (α):* This parameter specifies the probability of a link to being selected based on its pheromone value and also the impact of the gathered data by PANTs. The data influence and the exploitative nature of PANTs have increased by increasing the value of α. Decreasing the value of α will increase the PANTs' exploration and decrease the effect of pheromone value on the path selection procedure.
4) *Real-time information power (β):* The function of this parameter is very similar to α but the difference is that it controls the real-time information impact on the path selection procedure. This information is also gathered by VANTs.

Similar to ξ and λ, there should be a proper trade-off between α and β (i.e., $\alpha + \beta = 1$). The best condition occurs when $\alpha = 0.4$ and $\beta = 0.6$ in our simulation environment for AVCAF evaluation. Figure 6.3 illustrates the average travel time of the found paths by AVCAF as a function of the pheromone and real-time information. The average travel time converges towards two different values: 160s and 185s at the beginning (α from 0 to 0.2) and at the end (α from 0.8 to 1) of this diagram, respectively. This is because at the beginning, path finding is based more on vehicles real-time information ($0.8 \leq \beta \leq 1$) whereas at the end, it is based more on pheromone trial information ($0.8 \leq \alpha \leq 1$).

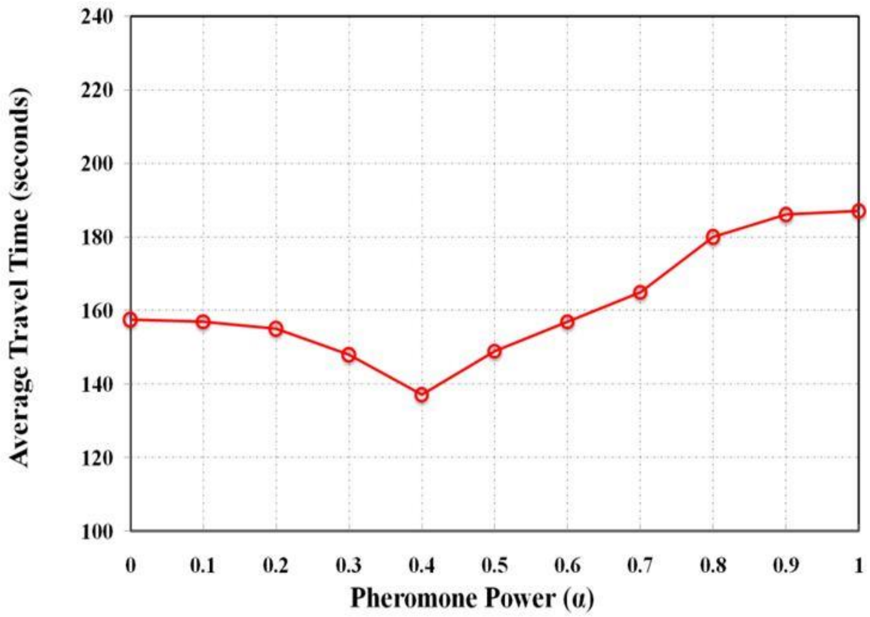

Figure 6.3. Average travel time for AVCAF as a function of pheromone power (λ = 0.6, ξ = 0.4, ρ = 0.5, N_a = 15, n = 3, γ = 5 TIs (50s))

5) *Pheromone evaporation rate (ρ):* Based on Di Caro (2004), this parameter plays an important role when there are multiple paths for selection and the characteristics of the environment change rapidly and dynamically. The described status is very similar to the vehicular environment which is the main focus of this thesis. Since ρ has a direct effect on having the proper trade-off between exploration and exploitation as well as the convergence speed of the algorithm, different values are examined for finding the best value of this parameter through the simulation and its result is demonstrated in Figure 6.4. At low values of ρ, the convergence speed is high because of the slow changes in the pheromone value of the links, while the algorithm does not converge at higher values of ρ because of the quick changes of pheromone trails on the links. The lowest average travel time happened when ρ = 0.3.

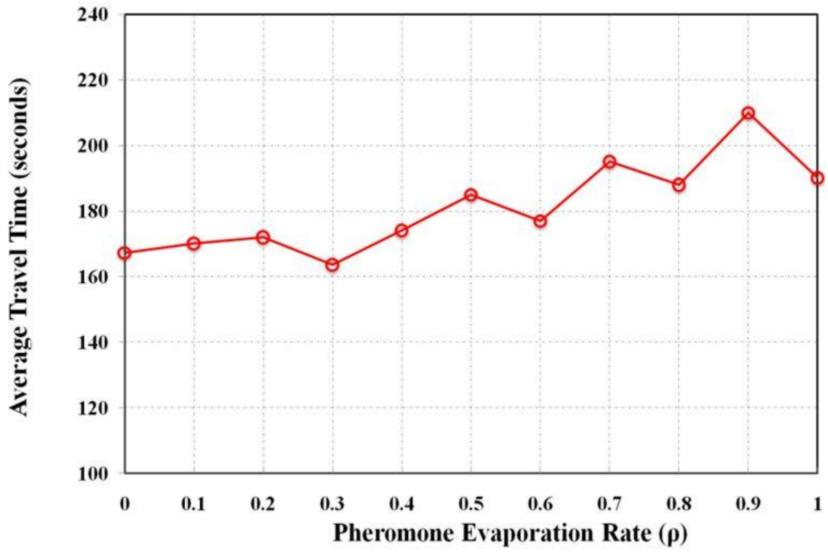

Figure 6.4. Average travel time for AVCAF as a function of pheromone evaporation rate (λ = 0.6, ξ = 0.4, α = 0.5, β = 0.5, N_a = 15, n = 3, γ = 5 TIs (50s))

6) *Number of alternative paths (n):* It is worth noting that although offering a large number of alternative paths for each OD pair allow better vehicle congestion avoidance and balancing, this leads to higher computational complexity. Moreover, since distance is one of the main metrics in AVCAF, large number of alternative paths which lead to computational overhead and long paths are not necessary. Selecting the proper value for this parameter can lead to decreasing both the average travel time and the computational cost. Since system response time is very critical criteria in a vehicular environments because of rapid changes, n = 3 was selected for AVCAF based on the results in Figure 6.5.

It means that AVCAF finds up to 3 alternative paths for each OD pairs in each segment. These three alternative paths are ordered based on the obtained probability value by Equation 4.4 (section 4.3.2.1) which encompasses various criteria (e.g. distance, capacity, density, travel time and speed) for each same OD pair. Consequently, the path with highest probability value has higher priority and a chance of being suggested to vehicles. For each OD pair, a First Come First Serve strategy is used in order to route the vehicles through these three alternative paths. The first 50% of routing requests are routed via the first path, i.e. the least congested shortest path. The next 33.4% of routing requests are routed through the second path which has lower priority than the first path but is still less congested compared with the other paths. The last 16.6% of routing requests are routed via the last path (3rd path). This routing cycle is continued for the coming routing requests. If AVCAF finds at most 2 alternative paths between specific OD pair, it routes 66.6% of vehicles through the first path and the other 33.4% is routed via the second path. However, all the vehicles are routed via the same path if there is only one path between specific OD pair. This last case is usually occurred when the vehicles are close to their destination. All of the above-mentioned percentages are obtained via Equations 4.10 and 4.11.

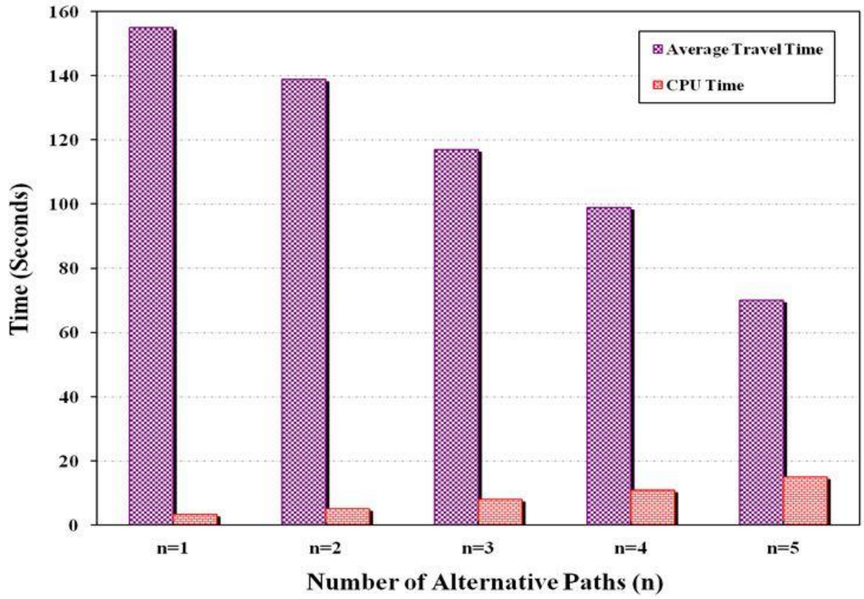

Figure 6.5. Average travel time for AVCAF as a function of number of alternative paths (λ = 0.6, ξ = 0.4, α = 0.5, β = 0.5, ρ = 0.5, N_a = 15, γ = 5 TIs (50s))

7) Number of ants (N_a) and re-generation period (γ): In AVCAF, the new path finding process is started periodically by regenerating a predefined number of new FANTs at predefined TIs. In general, a lower value for γ and a higher value for N_a lead to better average travel time and algorithm convergence speed, respectively. As a result, the computational cost and communication overhead of the system have increased. Figure 6.6 illustrates the average travel time for different values of γ.

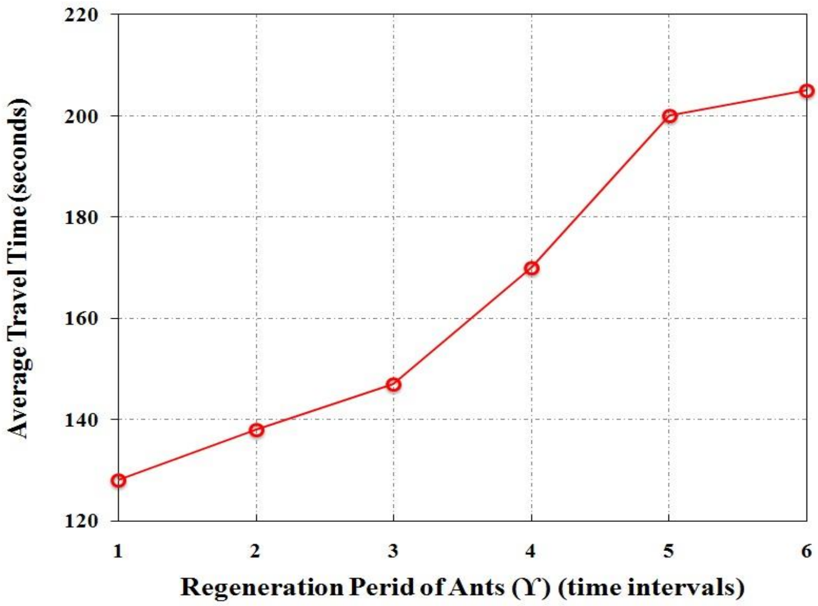

Figure 6.6. Average travel time for AVCAF as a function of the regeneration period of ants (λ = 0.6, ξ = 0.4, α = 0.4, β = 0.6, ρ = 0.3, n = 3)

Considering the trade-off between the average travel time on one side and the communication overhead and computational cost on the other side, 30 s or 3 TIs is selected as the regeneration period of FANTs in AVCAF. Moreover, the number of ants is chosen as a function of the number of destination nodes (*n-1*) in the segment and alternative paths (the number of nodes (source) in the segment × alternative paths). In our scenario, 15 ants (i.e. 3 (alternative paths) × 5 (destination nodes in the segment)) are put at each of the start point (source).The configuration parameters of AVCAF in NS-2 are summarized in Table 6.1.

Table 6.1. AVCAF configuration parameters and their values in NS-2

Parameter	Examined range	Value
ζ	0 - 1 (step: 0.1)	0.4
λ	0 - 1 (step: 0.1)	0.6
α	0 - 1 (step: 0.1)	0.4
β	0 - 1 (step: 0.1)	0.6
ρ	0 - 1 (step: 0.1)	0.3
Υ	1 TI – 6 TIs (step: 1 TI)	3 TIs or 30 s
N	1 - 5 (step: 1)	3 paths

6.2. Benchmark routing approaches

In this section an overview of the benchmark routing approaches for evaluating the performance of AVCAF along with the underlying reasons of their selection are represented. Dijkstra (Dijkstra, 1959), pure ant colony optimization (PACO) (Fan, Hua, Li, & Yuan, 2004) and HRS (Bogdan Tatomir & Rothkrantz, 2006) approaches are chosen for AVCAF performance evaluation. Each of these approaches is discussed in the following subsections.

6.2.1. Benchmark routing approaches

1) Dijkstra: This method is a greedy approach to solve the single source shortest problem. A greedy algorithm chooses the best local optimal choice at each stage with the hope of finding a global optimum solution. It repeatedly selects from the unselected vertices, vertex *v* nearest to source *s* and declares the distance to be the actual shortest distance from *s* to *v*. The edges of *v* are then checked to see if their destination can be reached by *v* followed by the relevant outgoing edges.

   Road map must be converted to a directed-weighted graph and the edges should be non-negative. If the edges are negative then the actual shortest path cannot be obtained. Dijkstra finds the distances from a given vertex *s* in the weighted digraph *D* = [*V*, *E*] to the rest of the vertices, where *V* and *E* are the set of vertices and edges, respectively.

   In the execution of Dijkstra algorithm, the vertex set of *D*, *V*, is partitioned into two sets, visited vertices, *P*, and unvisited vertices, *Q*. Moreover, a parameter δ_v is assigned to every vertex v ∈ V and indicates the distance between *s* and *v*. Initially all vertices are in Q. In the process of the algorithm, the vertices reachable from *s* move from *Q* to *P*. While a vertex *v* is in *Q*, the corresponding distance parameter, δ_v, is equal to infinity (∞). Once *v* moves to *P*, we have δ_v = *dist*(*s*, *v*). A formal description of Dijkstra algorithm is as follows:

   Dijkstra algorithm

   The parameter δ_v for every $v \in V$ such that δ_v = *dist*(*s*, *v*).

   1. Set $P := \emptyset$;; $Q := V$, δ_s:= 0 and δ_v:= ∞ for every $v \in V - s$:
   2. While *Q* is not empty do the following.

      Find a vertex $v \in Q$ such that $\delta_v = \min \{\delta_u : u \in Q\}$

      Set $Q := Q - v$, $P := P \cup v$.

      $\delta_u := \min\{\delta_u; \delta_v + c(v, u)\}$ for every $u \in Q \cap N^+(v)$.

The Dijkstra algorithm was selected as a benchmark algorithm, because it is a very popular algorithm and is used in most of the existing VTRSs for guiding the vehicles through the shortest paths.

2) PACO: In this approach, the proposed ACO algorithm by Dorigo (1992) which is discussed in Section 2 without any significant modification was used for the first time to find the shortest path between various OD pairs and this is why we call it pure ACO (PACO). Similar to the other ant-based approaches, PACO has 4 main steps, namely problem graph depiction, initialization, pheromone update and stopping procedure. Since ant-based algorithm is cornerstone of PACO approach and its concept is utilized in the AVCAF framework for path finding, PACO was chosen to verify and validate our proposed changes and enhancements. The comparison between PACO and AVCAF steps is summarized in Table 6.2 in order to clarify the differences between these two approaches.
3) HRS: This approach is an ant-based VTRS which splits road map into several smaller and less complex networks, named sector, by considering a hierarchy between the roads. HRS maintains a routing table for each intersection to route the vehicles. However, unlike AVCAF, this approach neglects the junctions in its routing mechanism which are one of the main points for mitigating the traffic congestion as well as vehicles re-routing. It is worth noting that HRS, similar to AVCAF, gathers the real-time traffic data from the traffic through the vehicles themselves. HRS includes two components, namely timetable updating system and route finding system. Timetable updating system computes the average travel time of each link on the road map through the gathered data by vehicles and sends this information to route finding system. This information will be used as input for an ant-based algorithm, called H-ABC (Tatomir & Rothkrantz, 2005), and embedded in route finding system. H-ABC utilizes ant-based algorithm to find the fastest way between

nodes of the road map considering streets/roads travel time. The HRS approach is compared with AVCAF because of their similarities in the use of segmentation and an ant-based algorithm for path finding and vehicle congestion reduction. The comparison between HRS and AVCAF steps is summarized in Table 6.2 in order to clarify the differences between these two approaches.

Table 6.2. Comparison among PACO, HRS and AVCAF considering their steps

Steps	PACO	HRS	AVCAF
Problem Graph Depiction	The problem graph is changed to a tree graph.	Road map is converted to a graph and divided into several smaller and less complex networks, named sector, by considering a hierarchy between the roads.	Segmentation and layering phases are used for problem graph preparation (see subsection 4.1)
Initialization	A number of FANTs (m) are located on the nodes and started to explore the problem graph by choosing the next node using probability function as follows: $P_{ij} = \frac{\tau_{ij} - \eta_{ij}{}^{\beta}}{\sum_{r \in A_k} \tau_{ir} \eta_{ir}{}^{\beta}}$, where $\eta_{ij} = 1/d_{ij}$, d_{ij} is the distance between nodes i and j. A_k is the reachable node set of ant k at node i.	Three types of ants are defined, namely local ants, FANTs and BANTs, to explore the road map. Local and FANTs are located on the nodes between and inside the sectors, respectively, and started to explore the problem graph by choosing the next node using probability value which is calculated by BANTs.	Two types of ant are defined, namely VANT and PANT, to consider historical and current traffic condition and also routing tables' update. PANTs are further divided into two FANTs and BANTs. A number of FANTs (Na) are located on the nodes and started to explore the problem graph by choosing the next node using probability function as follows: $P_{ij} = \frac{\alpha(\tau_{ij}) + \beta(\eta_{ij})}{\sum_{h \in \mathrm{tabu}_k} (\alpha(\tau_{il}) + \beta(\eta_{il}))} \times (\frac{1}{1 + \frac{1}{N_j}})$ $\eta_{ij} = (1 - D_{ij}) + \frac{PTS_{ij}}{100}$, Refer to subsection 4.2 for more details about above mentioned Equations and their variables.
Pheromone Update	The pheromone intensity is updated via BANTs while returning to the source nodes from destination nodes by using pheromone update rule as follows: $\tau_{ijd}^{new} = (1\text{-}\rho)\, \tau_{ijd}^{old} + \rho \Delta\tau_{ij}$, $\Delta\tau_{ij} = \begin{cases} \frac{1}{L} & if\ link\ (i,j) traversed\ by\ BANTs, \\ 0 & otherwise \end{cases}$ where $\rho \in$ (0,1] and L are pheromone evaporation ratio and average length of cycle, respectively.	The probability value is updated via BANTs while returning to the source nodes from destination nodes by using probability update rule as follows: $P_{ij}^{new} = P_{ij}^{old} + (1\text{-}\rho) \times (1 - P_{ij}^{old})$ where ρ is reinforcement value and obtained by: $\rho = \begin{cases} \frac{t_{ij}}{c\mu_d}, & \frac{t_{ij}}{c\mu_d} < 1 \\ 1 & otherwise \end{cases}$ Where t_{ij} and μ_d are travel time of link (i, j) and average delay, respectively. μ_d is obtained by: $\mu_d = \mu_d + \eta\ (t_{ij} - \mu_d)$ η is a weight to limit the influence of each delay on the average delay and is assigned to 0.1.	The pheromone intensity is updated via BANTs while returning to the source nodes from destination nodes by using pheromone update rule as follows: $\tau_{ijd}^{new} = (1 - \rho)\tau_{ijd}^{old} + \sum_{k=1}^{n} \Delta\tau_{ij}^{k}$, $\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{TT_{ij}^{k}} + \frac{1}{D_{ij}^{k}} + \frac{1}{LL_{ij}^{k}} & if\ the\ kth\ ant\ traversed\ link\ (i,j), \\ 0 & otherwise \end{cases}$ Refer to subsection 4.3 for more details about mentioned Equations and their variables.
Stopping Procedure	PACO is completed by reaching a predefined number of iterations, T.	HRS executes for an infinite number of cycles.	AVCAF executes for an infinite number of cycles. A cycle completed by reaching a predefined number of iterations where as an ant is dropped by arriving at a predefined maximum number of hops before reaching its destination and is set to n+1, where n is the number of nodes in a specific segment.

6.3. Performance evaluation

Three different scenarios, namely various vehicle densities, various system usage rates and accident condition, are implemented to evaluate AVCAF performance in reducing the vehicle traffic congestion. These scenarios are designed in order to evaluate AVCAF performance in both recurring and non-recurring congestion conditions. Besides, five different metrics, namely vehicles' average travel time, speed, distance, number of re-routings and number of congested roads, are defined as evaluation metrics. These five metrics are chosen due to their predominant role in both VTRSs and ant-based approaches. Each of these scenarios and evaluation metrics is discussed in the following subsections.

6.3.1. Simulation scenarios

Three different scenarios are implemented to evaluate AVCAF performance in reducing the vehicle traffic congestion in urban environments as follows:

1) Scenario with different vehicle densities: Both road map topology and vehicle density are key points in the simulation environments and they highly affect the accuracy of the obtained results. With regard to the road map topology, we used a complex real road map of Kuala Lumpur city (Figure 5.11) in our simulation to obtain closer results to real urban environments. In consideration of vehicle density, the performances of AVCAF, Dijkstra, PACO and HRS approaches were analyzed at different vehicle densities ranging from 100 to 1000 vehicles. As discussed in chapter 5, we have utilized TrafficModeler to generate vehicular traffic and movements. Various numbers of vehicles are generated and located on each of the desired origin points which are illustrated in Figure 6.7. The number of vehicles in each origin point varies from 25 to 250. In Figure 6.7, the circles on the left side of map represent these origins and the circle on the right side of the map is the vehicle's destination.

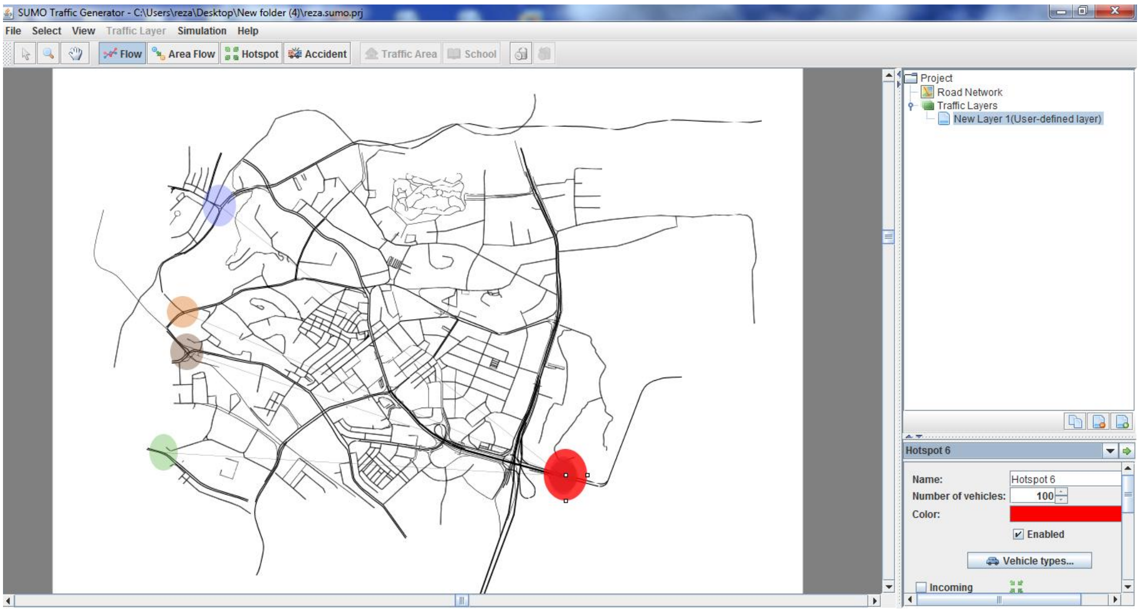

Figure 6.7. Scenario with various vehicle densities in TrafficModeler

2) Scenario with different driver usage rate: It cannot be assumed that in real world scenarios, every driver will follow the routing systems guidance. Therefore, in addition to vehicle densities, the usage rate (i.e. the proportion of drivers who use a specific guidance system and accept its guidance) and its impact on vehicle traffic congestion is also investigated in this thesis considering high traffic density (i.e. number of vehicles = 1000). In other words, 250 vehicles are located in each depicted origin point in Figure 6.7 and guided to destination point taking into account various system usage rates ranging from 10 to 100 percent. High traffic density is selected since vehicle traffic congestion is the main concern of this thesis. Moreover, in the most of existing VTRSs the same path is suggested to the drivers with same origin and destination, and consequently congestion will be switched from one route to another if a significant number of drivers utilize these system. Hence, AVCAF, Dijkstra, PACO and HRS approaches are compared with each other by considering different usage rates.

3) Scenario with accident condition: This scenario was simulated to evaluate the behavior of AVCAF when an accident takes place compared with the other four systems. This scenario is selected in order to evaluate the performance of AVCAF in

non-recurring congestion conditions. We split the vehicles in to four categories based on their chosen routing system, i.e. Dijkstra, PACO, HRS and AVCAF. We generated 100 vehicles for each category and dedicated 25 vehicles of each category to four starting points (i.e. circles in Figure 6.7) of our real road map. In other words, 400 vehicles, 100 vehicles for each category, are located at staring point of road map in our simulation environment. The total simulation period was 1000 seconds. An accident was generated at one of the main roads, which is depicted via azure across sign in Figure 6.8, after 300 seconds. This accident was omitted from the road at the 700th second of simulation. Vehicles may be forced to stop or halt on the lane for a defined time span by using the stop element in SUMO and this works similar to real accidents on the roads (Hrizi & Filali, 2010). However, we used accident feature of TrafficModeler to create an accident on the desired road.

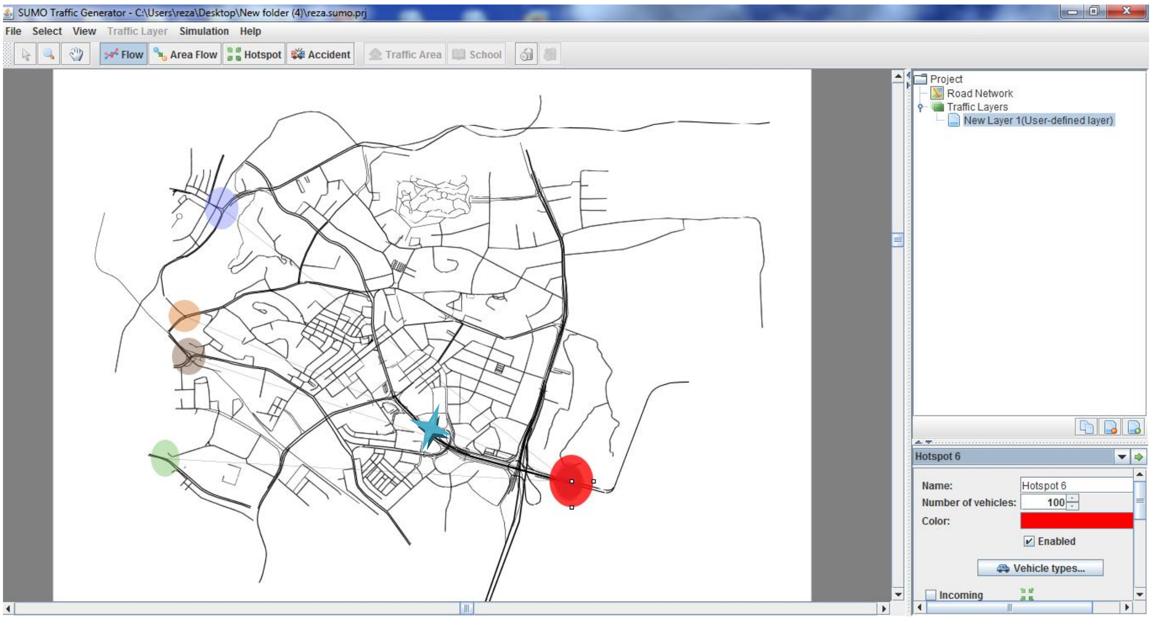

Figure 6.8. Accident scenario in TrafficModeler

6.3.2. Evaluation metrics

Five various evaluation metrics are defined and considered to evaluate AVCAF performance in reducing the vehicle traffic congestion in urban environments as follows:

1) Average travel time: it is a period of time spent by an individual vehicle or a group of vehicles to reach their destinations from their origins. The metric value is obtained from the sum of each individual vehicle spent travel time divided by the number of vehicles.
2) Average travel speed: it indicates an average speed of an individual vehicle or a group of vehicles during travel from origin to destinations. The metric value is obtained from the sum of each individual vehicle traveled speed divided by the number of vehicles.
3) Average travel distance: it indicates the mean distance passed by an individual vehicle or a group of vehicles to reach their destinations from their origins. The metric value is obtained from the sum of each individual vehicle traveled distance divided by the number of vehicles. This metric is used by most of the existing VTRSs to route the vehicles to their destinations.
4) Number of re-routings (Routing stability): The stability of ant-based approaches is related to their pheromone value stability. The degree to which the system remains in the same state can be examined by this metric. Pheromone value has direct impact on probability tables that constitute the basis for path finding procedure. At every node (e.g. intersection or junction), the node(s) with the highest (or higher) probability is selected as a next hop from the set of neighbors of the node. It is worth noting that a VTRS is stable as long as the highest probabilities and consequently pheromone values remain at the highest level for the entire time of simulation (Kroon, 2002). The system's stability can be measured by computing the number of re-routings. Number of re-routings counts the number of times that the highest probability switches to an alternative node. It is important that the re-routings frequency for a given vehicle during a trip stay low. From the driver point of view, changing the path to the destination too often can be distracting and annoying. From the system point of

view, having a low number of re-routings means decreasing the computational burden because the re-routing process is time consuming.

5) Number of congested roads: This metric indicates the number of congested roads (i.e. the road is assumed congested when average speed of the vehicles on that road is lower than 8.5 m/s) during the routing procedure. This metric is considered and evaluated since the vehicle traffic congestion reduction is the main concern of this thesis. Number of congested roads indicates whether a given system is able to mitigate the traffic congestion problem when there are both recurring and non-recurring congestion conditions. Since ant-based algorithms use stochastic search mechanism, pheromone update may deter ant-based approaches from finding an optimal solution. It is noteworthy to mention that the fine balance between exploration and exploitation is very important to the overall efficiency of ant-based approaches.

6.4. Simulation results for AVCAF evaluation

After investigating the effects of different parameters' values on the AVCAF and finding their proper values, the efficiency of AVCAF is evaluated in this section, by comparing with the discussed benchmark routing approaches considering the mentioned scenarios and evaluation metrics explained in Section 6.3. The obtained results are classified based on evaluation metrics and discussed in the following subsections.

6.4.1. Average travel time

This metric is calculated for each of the mentioned mechanisms (i.e. Dijkstra, PACO, HRS and AVCAF) and the results are illustrated in Figures 6.9, 6.10 and 6.11 for the three mentioned scenarios (i.e. various vehicle densities, system usage rate and accident condition), respectively. The obtained results for the first scenario (i.e. various vehicle densities), which are confirmed by Daganzo (1994), show that a vehicle's average travel time has a direct relationship with vehicle density: as the number of vehicles increases, the average travel time increases. However, this increment is too sharp in the Dijkstra and PACO systems versus other systems. This is because by using these two systems all of the vehicles with a same OD are guided through a same path (i.e. shortest path) without paying attention to other factors such as vehicle congestion, speed, density and accidents. In comparison, HRS and AVCAF improved the average travel time significantly. The average travel time at low vehicle densities (from 100 to 300) is almost the same for all systems, while this metric's value varied between each system at higher densities (from 400 to 1000). AVCAF has the best results in different vehicle densities and it decreased travel time up to 19%, 19% and 7% compared with Dijkstra, PACO and HRS, respectively. AVCAF improves the average travel time since it avoids congestion instead of recovering from it, which is not considered in other approaches. Moreover, in PACO and HRS, if many vehicles have the same OD pair at the same time, congestion can be transferred from one road to another. This problem is solved in AVCAF since it balances the traffic flow by using up to three alternative paths (*n*) at a same time.

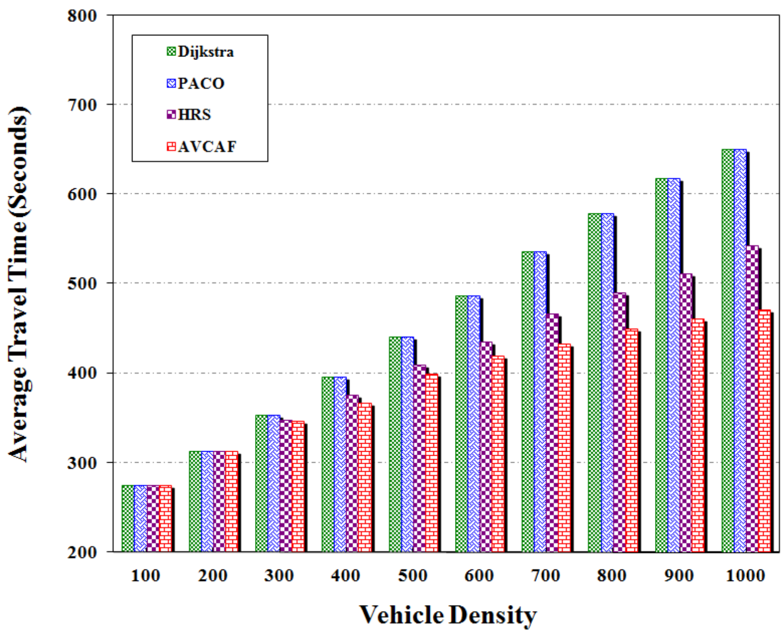

Figure 6.9. Average travel time for Dijkstra, PACO, HRS and AVCAF as a function of vehicle density

With regard to the drivers usage rate, average travel time of the vehicles that use Dijkstra and PACO systems has ascending trend for all usage rates based on the obtained results in Figure 6.10. However, in the case of HRS and AVCAF, this metric has descending trend at low usage rates whereas it has ascending trend at high usage rates. This is because HRS and AVCAF route the vehicle through the fastest path in time and less congested shortest paths, respectively. As a result, when the number of vehicles increases, more vehicles are routed via these paths and consequently the average travel time has increased. However, this increment for HRS is higher than AVCAF due to using only one path between various OD pairs.

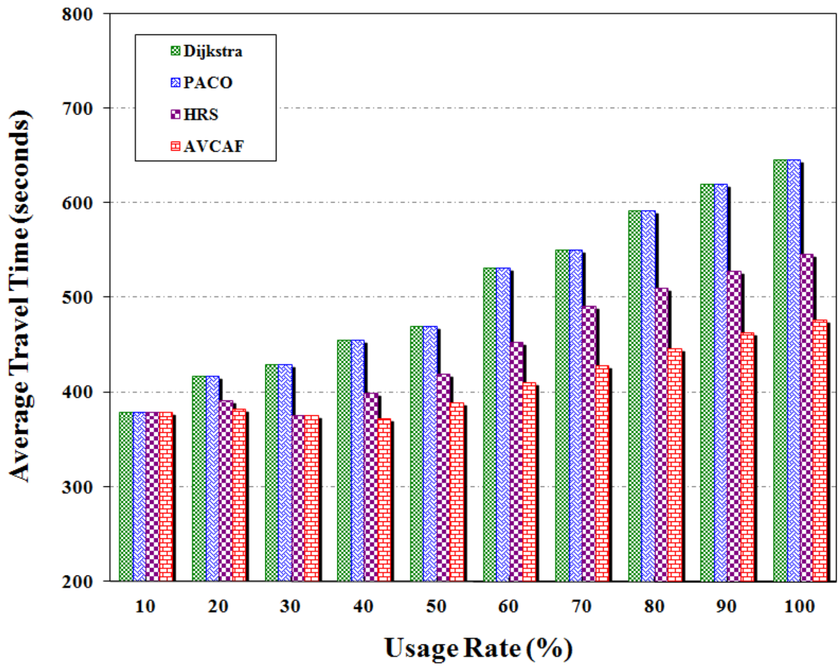

Figure 6.10. Average travel time for Dijkstra, PACO, HRS and AVCAF as a function of their usage rate

The necessary travel time for vehicles to reach their destination as a function of simulation step (i.e. simulation time) is counted for each system to measure their performance in facing non-recurring congestion conditions (e.g. accident) and is illustrated in Figure 6.11. As it can be seen, before the accident happens at the 300th second, all of the vehicles are guided through the shortest path via Dijkstra and PACO, fastest path in time via HRS, and up to three alternative least congested shortest paths via AVCAF and consequently the average travel time remained unchanged for these systems. However, after the accident and because of the occurred congestion on the link by the accident, HRS disabled (i.e. ignored) this road and started to reroute the vehicles through another alternative path without considering accidents severity, condition and duration. To solve this drawback and to consider accidents implicitly, AVCAF continued vehicles routing through the *n* alternative least congested shortest paths by considering road length, density, travel time and speed. This did not happen in Dijkstra and PACO due to the lack of attention to the dynamic changes of the vehicular environments. As a result, travel time started to increase sharply for the vehicles routed via the Dijkstra and PACO systems. In addition, travel time started to increase earlier for vehicles routed via HRS and AVCAF due to the use of prediction in these two systems. This increment for HRS is greater than AVCAF since HRS blocks the link with the accident, which is one of the main roads between OD pairs, while AVCAF reduces its chosen probability. From 400 to 700s of simulation, unlike Dijkstra and PACO, HRS and AVCAF rerouted the vehicles through the alternative paths and reduced the travel time value. At the 700th second when the accident is cleared from the road, the average travel time decreased rapidly for all of the systems and all the graphs smooth out to reach their initial values. AVCAF has the best reaction for non-recurring congestion since it uses travel time and vehicle density and travel speed prediction for vehicle routing and uses alternative paths from the beginning before congestion happens.

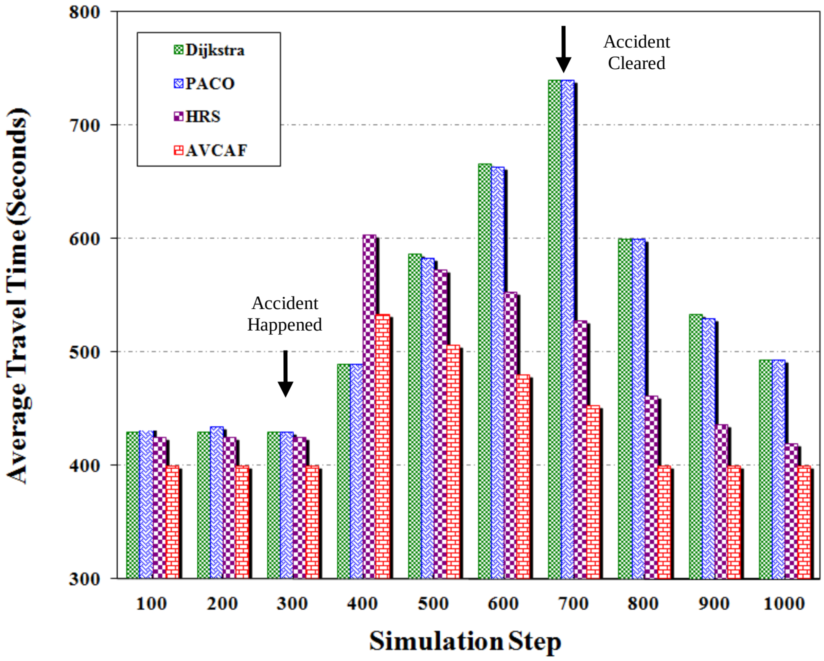

Figure 6.11. Average travel time for Dijkstra, PACO, HRS and AVCAF as a function of simulation step in accident condition

All in one, considering all scenarios, AVCAF outperforms the other approaches in terms of vehicles' average travel time at both recurring and non-recurring congestion conditions due to considering various metrics such as road length, density, travel time, speed in its routing mechanism and also routing vehicles via more than one alternative path (i.e. up to 3 paths).

6.4.2. Average travel speed

The relationship between the average travel speed and CO_2 emissions (and therefore, fuel consumption) is investigated by André and Hammarström (2000), and Barth and Boriboonsomsin (2009) and they discovered that there is a U-shape relationship between these metrics. This means that at a very low average travel speed, which normally occurs during vehicle congestion, and with a high number of stop-and-go driving events and extended engine idling on the road, the fuel consumption as well as CO_2 emissions increased by an average of 30% (Barth & Boriboonsomsin, 2008, 2009). Similarly, at very high speeds, the vehicles engine requires more power which leads to higher fuel consumption and more CO_2 emissions. Moderate average travel speeds ranging from 60 to 85 km/h (≈ 16.66 to 23.61 m/s) leads to the lowest fuel consumption and CO_2 emissions. The average travel speed of Dijkstra, PACO, HRS and AVCAF at various vehicle densities is illustrated in Figure 6.12. AVCAF obtained the best average speed rate at all vehicle densities by avoiding congestion and providing alternative paths before congestion occurred. By increasing the vehicle density, the average speed decreased smoothly from 24 to 16.7 m/s in AVCAF and as mentioned before, this speed range is the reported range for low fuel consumption and CO_2 emissions in the literature. The worst average travel speed and average travel time was generated by the Dijkstra algorithm and ranged from 25 to 10.5 m/s. The results for low vehicle densities (e.g. 100, 200 vehicles) were the same for all systems since the congestion level is low and all systems route the vehicles via a same path which is the shortest path.

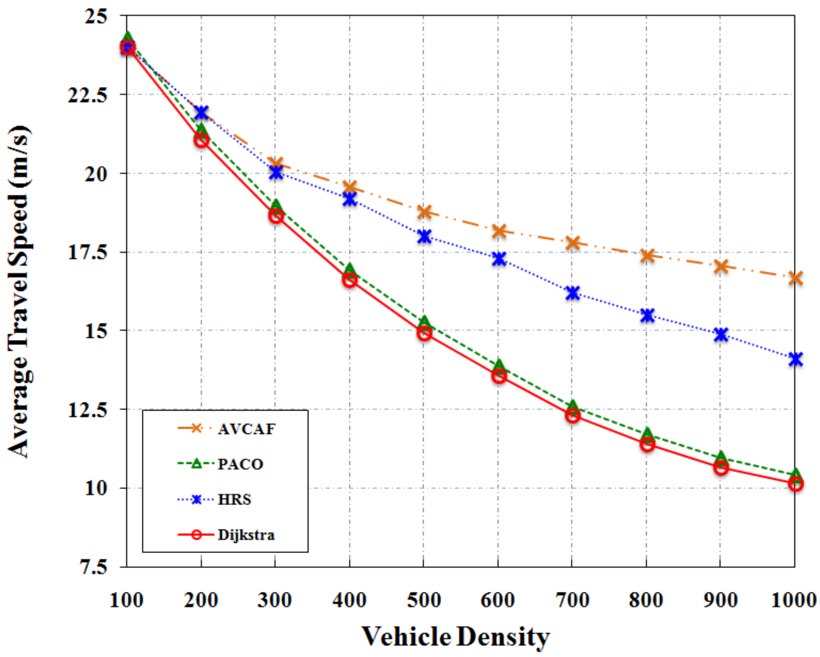

Figure 6.12. Average travel speed for Dijkstra, PACO, HRS and AVCAF as a function of vehicle density

Figure 6.13 presents the evaluation of average travel speed in all approaches under gradual increase of system usage rate. Based on the obtained results, AVCAF outperforms the other approaches at all usage rates due to distributing the vehicle traffic load through the *n* alternative paths. AVCAF and HRS has ascending trend at low usage rates since lower number of vehicles are routed through the alternative paths and the traffic load is distributed. However, at higher usage rates, higher number of vehicles is routed via alternative paths which lead to higher traffic load on these paths and consequently average travel speed is reduced. It is worth noting that this reduction is more significant in HRS due to routing the vehicles with same OD via the same path. Considering various usage rates, AVCAF preserves the average speed higher than 16.66 m/s which helps to reduce fuel consumption and CO_2 emission. Average travel speed of vehicles that use Dijkstra and PACO systems has an inverse relationship with usage rate: as the usage rate increases, the average travel time decreases due to sending all vehicles through the shortest path.

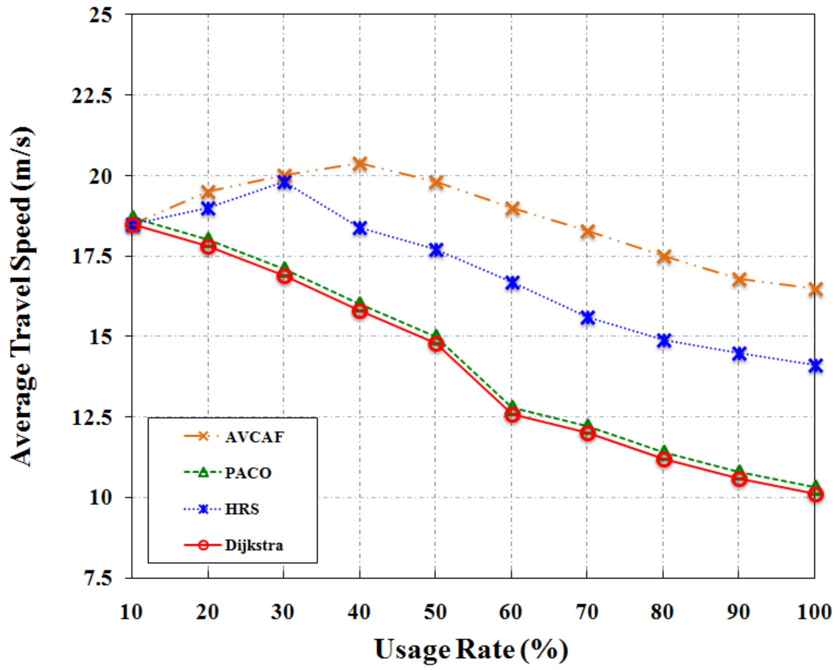

Figure 6.13. Average travel speed for Dijkstra, PACO, HRS and AVCAF as a function of their usage rate

With regard to the accident scenario, Dijkstra and PACO have the worst performance since they did not re-route the vehicles and routed them via the shortest path which is congested because of accident during the simulation time. After the accident take place at 300th second of simulation, the average travel speed is reduced for all approaches. However, this decrement is too sharp in the Dijkstra and PACO systems versus other systems due to the mentioned reason. HRS and AVCAF could alleviate this reduction by routing the vehicles through alternative path(s). AVCAF has the best performance due to considering various metrics such as road length, density, travel time, speed and also routing vehicles via more than one alternative path (i.e. up to 3 paths). By using AVCAF, vehicles' average travel speed is remained almost identical before and after accident which is one of its main advantages. It is worth noting that, similar to average travel time, AVCAF has the best performance in terms of average travel speed at all pre-defined scenarios.

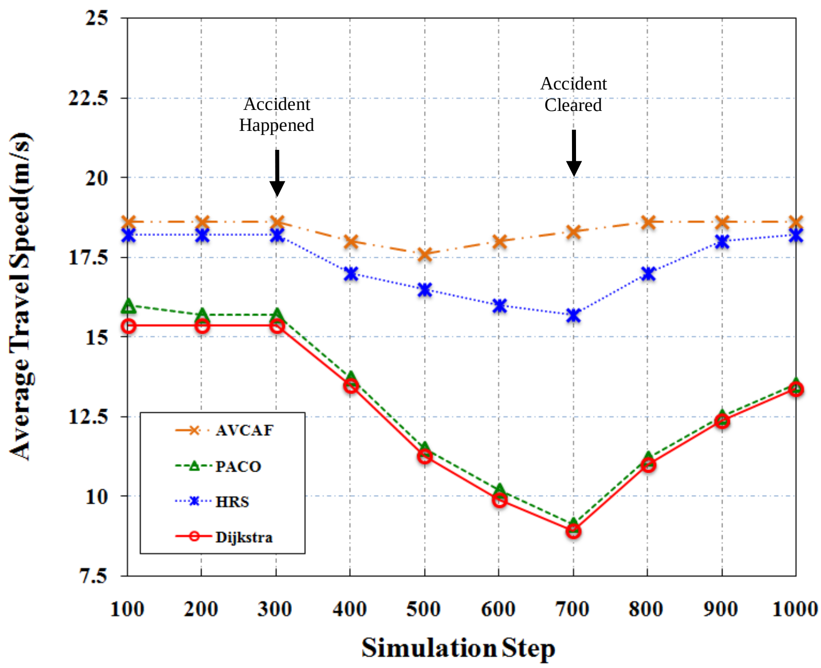

Figure 6.14. Average travel speed for Dijkstra, PACO, HRS and AVCAF as a function of simulation step in accident condition

6.4.3. Average travel distance

Figure 6.15 represents the average travel distance of vehicles that used Dijkstra, PACO, HRS and AVCAF systems for path finding under gradual rise of vehicle density. The worst average travel distance is associated with AVCAF and this is related to having a better average travel speed and time compared with the other systems. AVCAF mitigates vehicle traffic congestion problem by proposing slightly longer paths with less congestion instead of the shortest paths with congestion. This leads to an increase in the travel distance but a decrease in the travel time and increase in travel speed. It is worth noting that the average travel distance had increased at most 15% compared to the shortest path which was proposed by Dijkstra and PACO under the highest vehicle density (i.e., 1000 vehicles). Since AVCAF utilizes up to three alternative paths for avoiding and reducing vehicle congestion, its average travel distance is higher than Dijkstra, PACO and HRS which propose one alternative path and may transfer the congestion from one point to another point. Moreover, the average travel distance is constant and noted pendent on vehicle density, congestion or accidents and therefore is less suitable for vehicle congestion and avoidance systems. Moreover, the average travel distance of vehicles that use Dijkstra and PACO approaches is constant and is not dependent on vehicle density, congestion or accidents and therefore Dijkstra and PACO approaches are not suitable for vehicle congestion and avoidance systems.

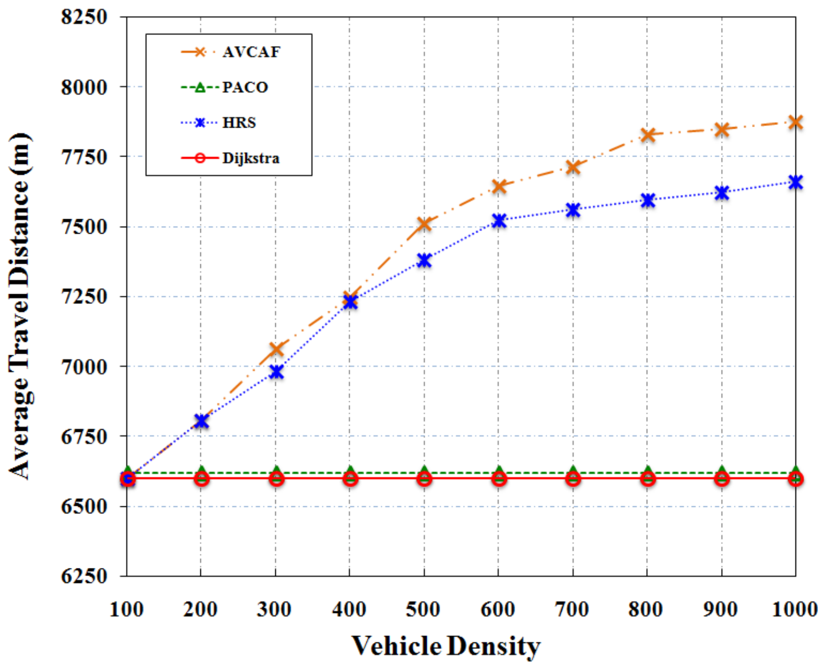

Figure 6.15. Average travel distance for Dijkstra, PACO, HRS and AVCAF as a function of vehicle density

Figure 6.16 depicts the average travel distance of vehicles in various usage rates of studied approaches. Unlike the average travel speed, AVCAF and Dijkstra had the worst and best performance for this evaluation metric under gradual increase of system usage rate. By using Dijkstra and PACO systems, average travel distance is reduced and converged to the shortest path as the usage rate increased. However, this metric increases for HRS and AVCAF as the usage rate increases. Higher average travel distance of AVCAF approach is related to its routing mechanism which route the vehicles through the paths with lower traffic and higher travel speed to reduce the vehicle traffic congestion problem. In most of the cases, these paths are longer than congested paths with shorter distances. In other words, although AVCAF increases the average travel distance, it reduces travel time by increasing travel speed which implicitly leads to lower fuel consumption and CO_2 emissions. As mentioned, since AVCAF utilize up to three alternative paths for avoiding and reducing vehicle congestion, their average travel distance is higher than Dijkstra, PACO and HRS which propose one alternative path and may transfer the congestion from one point to another point.

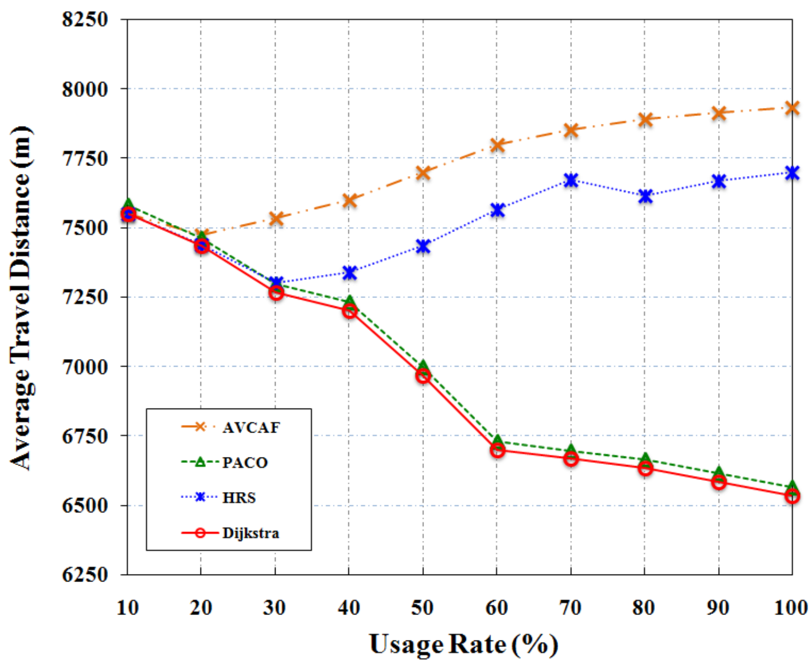

Figure 6.16. Average travel distance for Dijkstra, PACO, HRS and AVCAF as a function of their usage rate

With regard to the accident scenario, the lowest average travel distance is related to Dijkstra and PACO systems due to routing the vehicles through the shortest path without considering congestion conditions, especially non-recurring congestion (i.e., accidents) in this scenario. In the case of HRS and AVCAF, the average travel distance has increased dramatically after the accident take place at 300th second of simulation. However, this increase is more noticeable in the HRS system versus AVCAF due to its routing mechanism. That is to ignore the road with accident from routing procedure and to block it until the accident is removed. HRS and AVCAF could alleviate this increment by routing the vehicles through alternative path(s). AVCAF has better performance due to considering various metrics such as road length, density, travel time, speed and also routing vehicles via more than one alternative path (i.e. up to 3 paths). Although AVCAF increased the vehicles' average travel distance, it enhanced their average travel time and speed at both recurring and non-recurring congestion conditions.

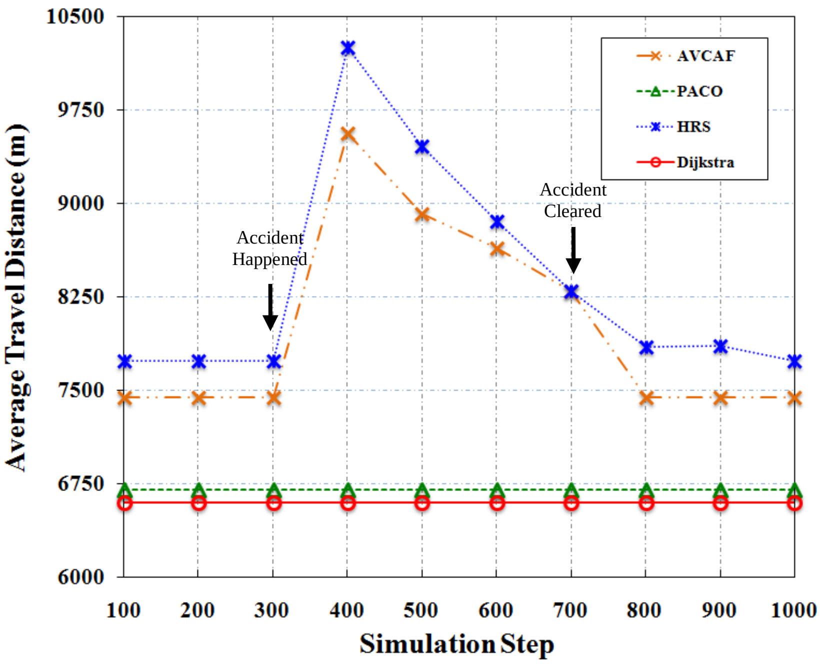

Figure 6.17. Average travel distance for Dijkstra, PACO, HRS and AVCAF as a function of simulation step in accident condition

6.4.4. Number of re-routings

Average number of re-routings is considered as routing stability of studied systems in this thesis. Figure 6.18 presents the comparison results of Dijkstra, PACO, HRS and AVCAF for average number of re-routings considering various vehicle densities. Since Dijkstra and PACO utilize only the distance parameter in their routing mechanism and this parameter has constant value, they route the vehicles through the shortest path and the number of re-routings is equal to zero. On the other hand, HRS and AVCAF re-route the vehicles during their travel from origin to destination based on the dynamic changes of vehicles traffic. However, AVCAF outperforms the HRS in terms of average number of re-routings due to considering various metrics such as road length, density, travel time, speed and also routing vehicles via more than one alternative path (i.e. up to 3 paths). In addition, the obtained results show that, similar to vehicle's average travel time, average number of re-routings has a direct relationship with vehicle density in the case of HRS and AVCAF approaches. In other words, as the number of vehicles increases, the average number or re-routings increases.

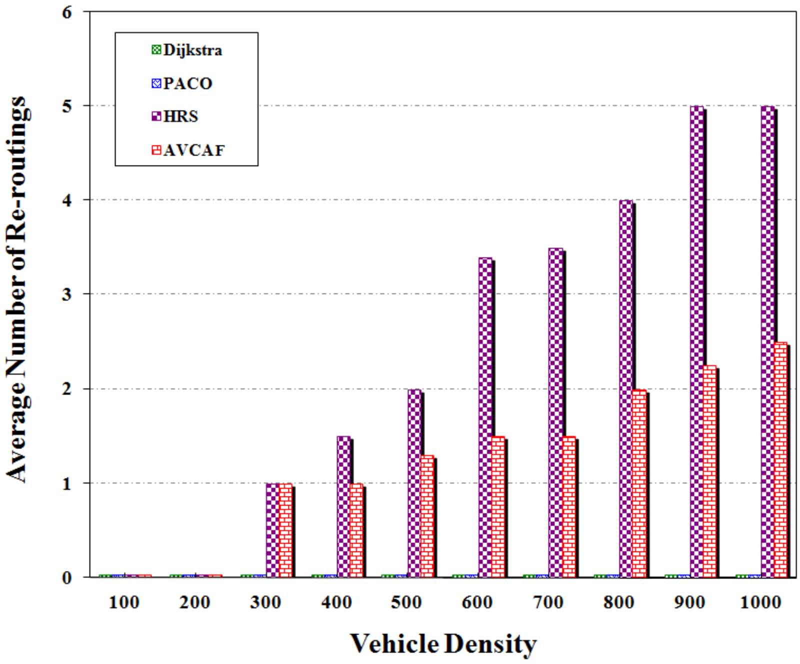

Figure 6.18. Average number of re-routings for Dijkstra, PACO, HRS and AVCAF as a function of vehicle density

Average number of re-routings for the studied approaches under the gradual increase of usage rate is depicted in Figure 6.19. The average number of re-routings is equal to zero in the case of both Dijkstra and PACO systems due to the fore mentioned reason. However, the average number or re-routings increases, as the usage rate increase for both HRS and AVCAF approaches. This is because as the usage rate increase, the number of vehicles that use these systems and routed through the same paths increases and consequently the number of re-routings has increased to mitigate the vehicle traffic congestion problem. It is worth noting that this increase is not noticeable in AVCAF approach due to routing the vehicles' with the same OD pair via up to three alternative paths from the early stages of routing procedure. This occurs such that the number of re-routing in AVCAF has decreased by 50% in comparison to HRS, in the highest usage rate condition.

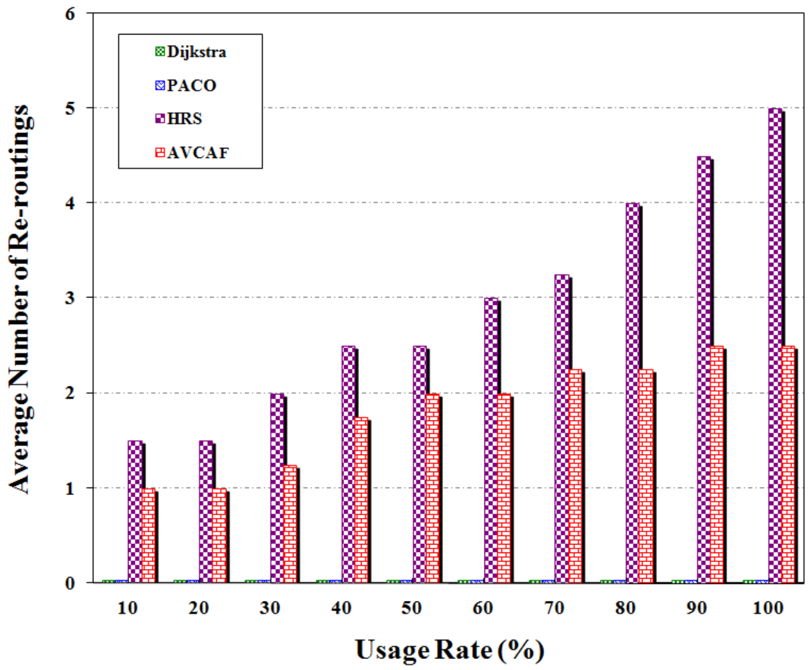

Figure 6.19. Average number of re-routings for Dijkstra, PACO, HRS and AVCAF as a function of their usage rate

Figure 6.20 represents the average number of re-routings for the vehicles that used Dijkstra, PACO, HRS and AVCAF systems for path finding under accident condition. Similar to the first and second scenarios (i.e. various vehicle densities and usage rates), the average number of re-routings for Dijkstra and PACO is zero due to considering the distance as the only and main parameter in their routing mechanism. Before accident, the average number of re-routings is constant for both AVCAF and HRS systems. Although this metric has increased for both of these systems after the accident take place at 300th second of simulation, it has constant and ascending trend for AVCAF and HRS systems, respectively, during the accident condition. This is because of three following reasons: 1) AVCAF uses various metrics such as road length, density, travel time, speed in its path finding procedure, while, HRS uses only road travel time, 2) AVCAF routes the vehicles with the same OD pair via more than one alternative path (i.e. up to 3 paths), while, HRS uses only one path, and 3) HRS disables (i.e. ignores) the road where an accident has happened during the accident, while, AVCAF decreases that roads probability value based on the accident severity (i.e. this is considered by using various metrics such as road's number of lanes, density, travel time and speed). Considering all of the scenarios, Dijkstra and PACO ignore both recurring and non-recurring congestion conditions in their routing procedure and their average number of re-routings is *zero*, while, AVCAF considers both of these conditions in its routing and routes the vehicles through the less congested shortest paths along with the lowest number of re-routings.

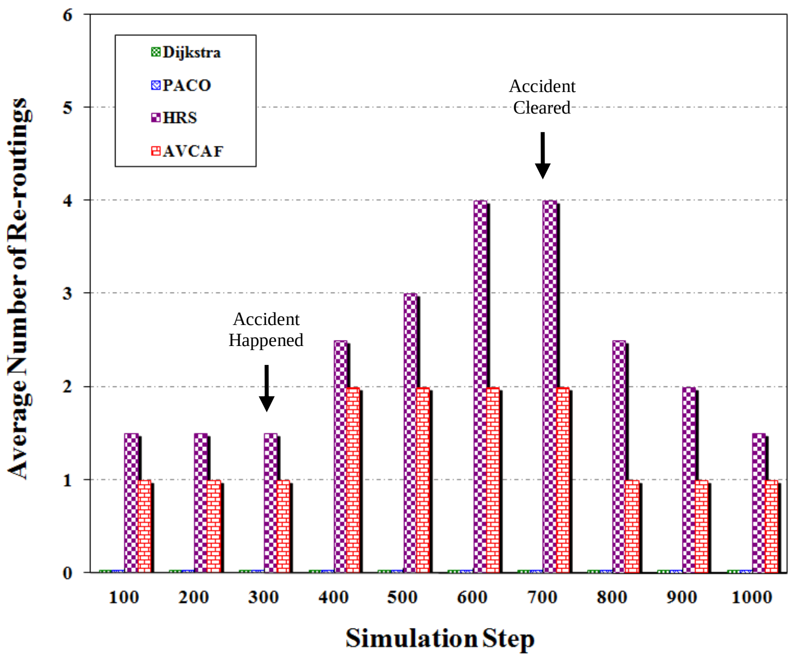

Figure 6.20. Average number of re-routings for Dijkstra, PACO, HRS and AVCAF as a function of simulation step in accident condition

6.4.5. Number of congested roads

Another parameter which is prominently important for performance evaluation of any VTRS is the number of congested roads. The optimal usage of the existing roads and streets capacity, which is the main reason of VTRSs emersion, can be examined via this parameter. Accordingly, the performance of Dijkstra, PACO, HRS and AVCAF in terms of number of congested roads under gradual rise of vehicle density is evaluated and the obtained results are depicted in Figure 6.21. As the number of vehicles increases, the number of congested roads increases for all approaches. In this evaluation the AVCAF outperforms the other approaches and routes the vehicles in a way that the lowest number of congested roads is created in various vehicle densities. Using up to three alternative paths for guiding the vehicles with the same OD pair is the main reason of this phenomenon. Routing the vehicles through the shortest path without considering traffic conditions is the main reason of worst performance of Dijkstra and PACO systems.

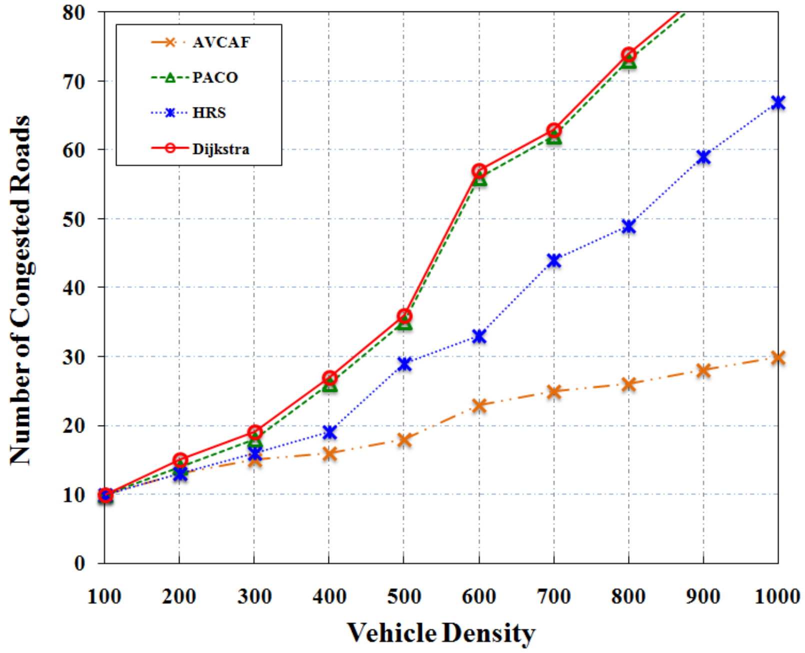

Figure 6.21. Average number of congested roads for Dijkstra, PACO, HRS and AVCAF as a function of vehicle density

Figure 6.22 represents the evaluation comparison of the number of congested roads in all approaches by applying various system usage rates. Dijkstra and PACO have ascending trend for all usage rates. However, in the case of HRS, this metric has descending trend at low usage rates whereas it has ascending trend at high usage rates. This is because HRS routes the vehicles with the same OD pair through only one path. As a result, when the number of vehicles increases, more vehicles are routed via these paths and consequently the congestion is transferred from one point to another point and the number of congested routes has increased. In the case of AVCAF, number of congested roads has inverse relationship with system usage rate. This is due to following reason: as the number of vehicles that use AVCAF for path finding increases, the traffic congestion is handled in a better way and distributed throughout the existing paths due to using up to three alternative paths in its routing procedure.

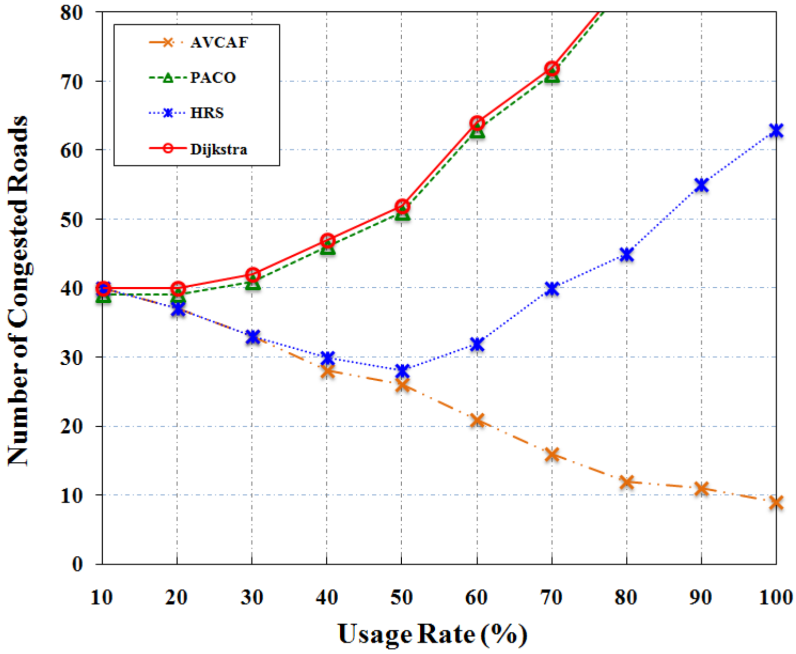

Figure 6.22. Average number of congested roads for Dijkstra, PACO, HRS and AVCAF as a function of their usage rate

The number of congested roads for the benchmark routing systems and AVCAF in accident condition is represented in Figure 6.23. As it can be seen, the number of congested roads is constant for all approaches. However, there is ascending trend for all approaches after the accident take place at 300th second of simulation. However, this increase is not significant in the case of AVCAF due to its unique routing mechanism that uses up to three alternative paths from the early stages of the routing procedure. Considering the other approaches, HRS outperforms the Dijkstra and PACO due to its re-routing mechanism. As expected, Dijkstra and PACO have the worst performance during the accident condition because of insisting on sending the vehicles through the shortest path without considering recurring and non-recurring congestion conditions. When the accident is removed from the street, HRS and AVCAF reduce the number of congested roads faster than Dijkstra and PACO approaches.

Similar to the average number of re-routings, AVCAF outperforms the other approaches in terms of number of congested roads considering both recurring and non-recurring congestion conditions. As a result, the optimal usage of existing roads and streets capacity is obtained via AVCAF framework.

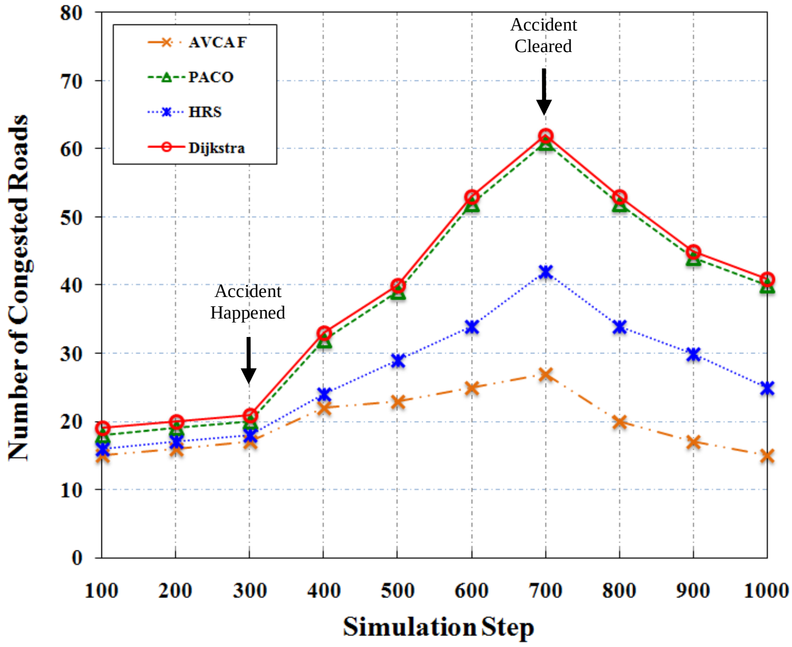

Figure 6.23. Average number of congested roads for Dijkstra, PACO, HRS and AVCAF as a function of simulation step in accident condition

6.5. Conclusion

This chapter has presented the evaluation results retrieved from running of implemented simulation by taking advantage of the line charts and bar chart, in order to facilitate the performance comparisons. Since the performance of ant-based approaches is highly dependent to the value of their various parameters, the proper value for various AVCAF parameters is found by examining different values for them through the simulation. The performance evaluation in this study is performed in comparison with the Dijkstra, PACO and HRS approaches. The selection reason(s) of each of these benchmark routing systems are also discussed in this chapter. The consideration of evaluation metrics and scenarios for comparison and assessment of our approach is performed in accordance with objectives of the study. Three different scenarios were defined in order to evaluate the performance of our approach in both recurring and non-recurring congestion conditions. The obtained results show that AVCAF outperforms the other approaches in terms of average travel time, speed, number of re-routings and number of congested roads in both recurring and non-recurring congestion conditions. This is due to considering various metrics such as road length, density, travel time, speed in its path finding procedure and also routing vehicles via more than one alternative path (i.e. up to 3 paths). Lower average travel time and higher average travel speed prove that AVCAF routes the vehicles through the less congested shortest paths, meanwhile, lower number of re-routings and congested roads prove that AVCAF distributes the vehicle traffic in order to achieve vehicle traffic congestion mitigation and optimal usage of the existing roads and streets capacity. However, AVCAF had higher average travel distance compared with others, especially in recurring congestion conditions (i.e. various vehicle densities and system usage rates), which can be ignorable due to its better average travel time and travel speed. The results also proved that ant-based algorithm (i.e. PACO) can find the shortest path by considering distance as its main metric which is proved mathematically by Shah et al. (2013). The major conclusions are drawn in next chapter.

CHAPTER 7: CONCLUSION AND FUTURE WORK

This chapter concludes this thesis. It includes an overview of the problem statement, research purpose, reached goals and findings. Moreover, all of the main points and obtained results are represented. Finally, this chapter concludes by discussing open issues in ant-based VTRSs and proposing some directions as a future work section.

7.1. Overview

Over the last decade, vehicle population has dramatically increased all over the world. This large number of vehicles coupled with the limited capacity of roads and highways lead to heavy traffic congestion. Besides, it gives rise to air pollution, driver frustration, and costs billions of dollars annually in fuel consumption. VTRSs are reported as an effective solution with reasonable cost for congestion mitigation in recent years. Based on our discussion of the existing approaches and related literature that have been proposed by other researcher, bio-inspired-based VTRSs, especially ant-based VTRSs, have been reported as promising solution for vehicle traffic routing and congestion problems. Although several ant-based VTRSs are developed over the years, there is not any framework for applying ant-based algorithms to VTRSs. In addition, non-recurring congestion conditions are not considered in most of the existing approaches due to the complexity of these unpredictable events. Moreover, drivers with the same origin and destination are routed through a same path by existing VTRSs. As a result, when a significant number of drivers utilize these systems (i.e., system usage rate is high), the congestion will be transferred from one route to another. Therefore, the main goal of this thesis was designing a vehicle traffic routing framework that avoids and mitigates both recurring and non-recurring congestion conditions by providing least congested shortest paths as alternative routes for drivers by taking advantage of VANET and bio-inspired approaches (e.g. ant-based algorithm). Five objectives are defined to achieve the main goal of this thesis. These objectives and the way of fulfilling them are discussed in the following section.

7.2. Reached Objectives

Five objectives are defined in Chapter 1 in order to achieve the mentioned main goal of this thesis as follows: 1) To present a classification for the existing research trends within the area of vehicle traffic routing and congestion control, 2) To provide a taxonomy and statistical overview of ant-based approaches, which are used in vehicle traffic routing systems, 3) To design a scalable vehicle traffic congestion avoidance framework using ant-based algorithm, 4) To develop the proposed framework in simulation environment using network simulator, mobility generator and VANET simulator, 5) To evaluate and analyze the proposed framework with different set of scenarios and evaluation metrics. In the following paragraphs, the reached objectives of this thesis are briefly discussed on an objective by objective basis.

- **Presenting a classification for the existing research trends within the area of research**

Indeed reviewing the existing research trend within the area of vehicle traffic routing and congestion mitigation, which was conducted in the first part of Chapter 3, helped us to achieve adequate theoretical knowledge to finalize the objectives of our thesis. By reviewing the existing approaches, we found that the most important distinction among VTRSs is whether the system utilizes a static or a dynamic routing algorithm for vehicle routing. This finding leads to a classification of VTRSs, namely static and dynamic VTRSs. However, static approaches are not suitable for vehicle congestion mitigation due to lack of considering real-time traffic information. Dynamic VTRSs that uses real-time traffic information in their routing algorithm are proposed to overcome static VTRSs' problems. After extensive overview of dynamic VTRSs, we found ant-based algorithms are promising solutions for vehicle routing and congestion avoidance. Hence, the second objective is defined to provide a taxonomy and statistical overview of ant-based approaches in the field of our research.

- **Providing a taxonomy and statistical overview of ant-based approaches within the area of research**

The second objective was reached in the second part of Chapter 3. The literature in the area of bio-inspired-based VTRSs is extensive. However, based on our investigation, ant, bee, genetic and PSO are most commonly used bio-inspired algorithms among researchers over the years in the field of VTRSs. Among these bio-inspired algorithms, the use of ant-based algorithms has been reported as promising and one of the best approaches for congestion control and traffic management in many research studies due to their better adaptability and processing time. After providing an overview of bio-inspired-based (i.e. bee, genetic and PSO) VTRSs in Section 3.2.1, a novel taxonomy and statistical overview of ant-based VTRSs is provided in Sections 3.2.2 and 3.2.2.4, respectively (i.e. which was our second objective). The major characteristics of ant-based VTRSs are also discussed in this chapter. Based on our discussion of the existing approaches and related literature that have been proposed by other researcher, we found that there is not any framework for applying ant-based algorithms to VTRSs. Moreover, non-recurring congestion conditions are not considered in most of the existing approaches due to the complexity of these unpredictable events. Hence, we have decided to propose a framework to cover these gaps.

- **Designing a scalable Ant-based Vehicle Congestion Avoidance Framework (AVCAF)**

The third objective, namely designing a scalable framework that uses ant-based routing algorithm in order to route the vehicles in a way that avoids the congestion by considering both recurring and non-recurring congestion conditions in its routing mechanism was reached in Chapter 4. Ant-based algorithm proactively reroute the vehicles when there are signs of congestion and in this way avoids the congestion occurrence instead of recovering from it. Initialization, optimal path finding, and optimal path suggestion are the three phases involved in AVCAF. Layered and segmented model was introduced and used in initialization phase in order to solve the scalability problem in ant-based algorithms, decrease the computation time of the routing algorithm and mitigate the drawbacks of centralized architectures via a distributed centralized architecture. Combination of historical and real-time traffic data, obtained via VANET infrastructure, was used to predict the travel speed of each road. Ant-like agents were designed to find the least congested shortest paths as alternative routes to the drivers by considering various metrics (i.e. vehicles' travel speed, travel time, density, road length and width). The optimal paths are suggested to drivers via VANET infrastructure on a junction to junction basis.

- **Developing the proposed framework via simulation tools**

Since AVCAF includes VANETs for both real-time data gathering and optimal paths suggestion, its implementation, evaluation and testing in real world involve high cost and in most of the cases impossible task if metrics such as testing area, mobility and number of vehicles are taken into account. Besides, most experiments are not repeatable and require high cost and efforts. Therefore, simulation tools and environments were selected and used to AVCAF development and implementation. After extensive investigation on simulation tools, TraNS, SUMO and NS-2 are selected as VANET simulator, mobility generator and network simulator for this purpose. The implementation details of AVCAF were covered in Chapter 5 on a phase to phase basis. Schematic simulation framework, snapshots from simulation tools and pseudo code based description were used to provide more details about the AVCAF implementation.

- **AVCAF evaluation and analyze**

Our last objective was achieved in Chapter 6 by comparing its performance with three other approaches, namely Dijkstra, PACO, HRS. One of the main concerns of evaluation, which was to perform the evaluation under realistic scenarios, was mitigated by choosing a real road map in Chapter 5 for evaluation purpose. AVCAF includes various parameters such as α, β, ζ, λ and ρ in its routing procedure and its performance is highly dependent to the value of these parameters. Hence, at the first part of Chapter 6, the proper value for these parameters was found by examining various values for them through the simulation. In order to evaluate AVCAF performance in both recurring and non-recurring congestion conditions, three different scenarios, namely various vehicle densities, various system usage rates and accident condition, were designed. Besides, five different metrics, namely vehicles' average travel time, speed, distance, number of re-routings and number of congested roads, are defined as evaluation metrics. The obtained results proved that AVCAF outperforms the other approaches in terms of average travel time, average travel speed, number of re-routings and number of congested roads in both recurring and non-recurring congestion conditions. The main findings and contributions of this thesis are discussed in the following section.

7.3. Findings and Contributions

Generally, providing an overall view of road map network along with existing traffic condition can be useful for travelers who want to travel through the optimal paths. However, the way of using and managing these data to find an optimal path is more important. Extensive simulation-based experiments were conducted in this thesis for AVCAF performance evaluation.

The experimental results proved that the proposed framework (i.e. AVCAF) along with its ant-based routing algorithm is very effective in avoiding and mitigating congestion and has high adaptability with dynamic and quick changes of vehicular traffic and environments. Based on the obtained results, AVCAF outperforms the other approaches in terms of vehicles' average travel time and speed at both recurring and non-recurring congestion conditions (see Sections 6.4.1 and 6.4.2) due to considering various metrics such as roads' length and width, vehicles density, travel time and speed in its routing mechanism and also routing vehicles via more than one alternative path (i.e. up to 3 paths). Although AVCAF routes the vehicles via slightly longer paths compared with other approaches in both various vehicle densities and system usage rates scenarios (see Figures 6.15 and 5.16), it can be condoned due to its better travel time and speed. It is worth noting that AVCAF outperforms the HRS approach in term of average travel distance in the accident scenario (see Figure 6.17).

In another experiment, average number of re-routings or routing stability was analyzed. This also showed that AVCAF, considering all scenarios, has better routing stability due to routing vehicles via alternative paths by considering various metrics from the early stages of routing procedure (see Section 6.4.4). In addition, the optimal usage of the existing roads and streets capacity, which is the main reason of VTRSs emersion, was examined via another parameter, called number of congested roads. In this evaluation, the AVCAF outperforms the other approaches by rerouting the vehicles when congestion is sensed by designed ant agents. As a result, the vehicles are routed in a way that the lowest number of congested roads is created in various vehicle densities (see Section 6.4.5). It is worth noting that the results further showed that the ant algorithm (i.e. PACO) can find the shortest path by considering distance as its main metric which was also proved mathematically.

As a conclusion, scalability which is one of the main problems of ant-based approaches was solved by proposing layered and segmented model. In addition, the results of accident scenario proved that the non-recurring congestion conditions were handled efficiently along with the recurring congestion conditions by AVCAF. Moreover, AVCAF remains effective and efficient even with high number of simultaneous users (i.e. high system usage rates).

The main findings and contributions of this thesis are published in conferences and journals and also a copyright as follows:

- Mohammad Reza Jabbarpour, Hossein Malakooti, Ali Jalooli and Rafidah Md Noor, "A Dynamic Vehicular Traffic Control Using Ant Colony and Traffic Light Optimization", *International Conference on Systems Science (ICSS 2013),* September 10-12, 2013, Wroclaw, Poland.
- Mohammad Reza Jabbarpour, Hossein Malakooti, Rafidah Md Noor, Nor Badrul Anuar and Norazlina Khamis, "Ant Colony Optimisation for Vehicle Traffic Systems: Applications and Challenges", *International Journal of Bio-Inspired Computation (IF = 1.682, Q1)*, 2014, Vol. 6, No. 1, pp. 32-56.
- Mohammad Reza Jabbarpour, Ali Jalooli, Erfan Shaghaghi, Rafidah Md Noor, Leon Rothkrantz, Rashid Hafeez Khokhar and Nor Badrul Anuar, "Ant-based Vehicle Congestion Avoidance System using Vehicular Networks", *Engineering Applications of Artificial Intelligence (IF = 1.962, Q1),* 2014, Vol. 36, pp. 303-319.
- Mohammad Reza Jabbarpour and Rafidah Md Noor, "A Novel Green Car Navigation System for Vehicle Congestion Avoidance using Ant-based Algorithm", *Copyright*, 2014, No: LY2014001218.
- Mohammad Reza Jabbarpour, Rafidah Md Noor and Rashid Hafeez Khokhar, "Green Vehicle Traffic Routing System using Ant Colonial-based Algorithm", *Journal of Network and Computer Applications (IF = 1.772, Q1)*, 2014, (Under review, Submitted Nov. 2014)

Some other achievements accomplished during doing my PhD including patent, journal and conference papers are listed as follows:

<u>*Patent:*</u>

- Mohammad Reza Jabbarpour and Rafidah Md Noor, "System and Method For Relieving Traffic Congestion", *Patent*, 2014, No: PI 2014703373.

<u>*Journal Papers:*</u>

- Mohammad Reza Jabbarpour, Rafidah Md Noor, Rashid Hafeez Khokhar and Chih-Heng Ke, "Cross-layer Congestion Control Model for Urban Vehicular Environments", *Journal of Network and Computer Applications (IF = 1.772, Q1)*, 2014, Vol. 44, pp. 1-16.
- Hossein Malakooti, Ali Jalooli, Mohammad Reza Jabbarpour, Ali Reza Marefat and Rafidah Md Noor, "Opportunistic XOR-Coding Content Propagation in VANET", *International Journal of Computer & Electrical Engineering*, 2014, Vol. 6.
- Ali Jalooli, Erfan Shaghaghi, Mohammad Reza Jabbarpour, Rafidah Md Noor, Hwasoo Yeo, and Jason J. Jung, "Intelligent Advisory Speed Limit Dedication in highway using VANET", *Scientific World Journal (IF = 1.7, Q1)*, 2014, Vol. 2014.
- Mohammad Reza Jabbarpour, Alireza Marefat, Ali Jalooli, Rafidah Md Noor, Rashid Hafeez Khokhar and Jaime Lloret, "Performance Analysis of V2V Dynamic Anchor Position-based Routing Protocols", *Wireless Networks Journal (IF= 1.055, Q3)*, 2014, DOI: 10.1007/s11276-014-0825-8.
- Mohammad Reza Jabbarpour, Ali Jalooli, Erfan Shaghaghi, Alireza Marefat, Rafidah Md Noor and Jason J. Jung, "Analyzing the Impacts of Velocity and Density on Intelligent Position-based Routing Protocols", *Journal of Computational Science (IF=1.567, Q2)*, 2014, DOI: 10.1016/j.jocs.2014.11.003
- Mohammad Reza Jabbarpour, Rafidah Md Noor and Saied Ghahremani, "Dynamic Congestion Control Algorithm for Vehicular Ad-hoc Networks", *International Journal of Software Engineering and Its Applications (IJSEIA)*, 2013, Vol. 7, No. 3.
- Erfan Shaghaghi, Mohammad Reza Jabbarpour, Rafidah Md Noor and Hwasoo Yeo, "Adaptive Green Traffic Signal Controlling using Vehicular Communication", *Journal of Network and Computer Applications (IF=1.772, Q1)*, 2015. (Under review)

Conference Papers:

- Rafidah Md Noor, Rashid H. Khokhar, Mohammad Reza Jabbarpour, Sajad Khorsandroo, Norazlina Khamis and Oche Michael, "Using VANET to Support Green Vehicle Communications for Urban Operation Rescue", *12th International Conference on ITS Telecommunications*, November 5-8, 2012, Taipei, Taiwan.
- Saeid Ghahremani, Mohammad Reza Jabbarpour, Sajad Khorsandroo, Mohamed Ahmed and Rafidah Md Noor, "A Traffic Control Model on VANET Environment for Minimum Road Risk in a Shortest Way", *The International Conference on Green High Performance Computing (ICGHPC'13)*, March 14-15, 2013, St.Xavier's Catholic College of Engineering, India.
- Mohamed Ahmed, Mohammad Reza Jabbarpour, Mostofa Kamal Nasir,Saeid Ghahremani, Sajad Khorsandroo, Syed Adeel Shah Ali, and Rafidah Md Noor, "Vehicle Adhoc Sensor Network Framework to Provide Green Communication for Urban Operation Rescue", *3rd International Conference on Information and Network Technology (ICINT 2013)*, April 1-2, 2013, Singapore.
- Mohammad Reza Jabbarpour, Hossein Malakooti, Masumeh Taheri and Rafidah Md Noor, "The Comparative Analysis of Velocity and Density in VANET Using Prediction-Based Intelligent Routing Algorithms", *The 2nd International Conference on Future Generation Communication Technologies (FGCT 2013)*, December 12-14, 2013, London, UK.
- Ali Jalooli, Mohammad Reza Jabbarpour, Hossein Malakooti, Erfan Shaghaghi, Alireza Marefat, and Rafidah Md Noor, "Adaptive advisory speed limit dissemination using vehicular communication", *Asia Pacific Conference on Wireless and Mobile 2014 (APWiMob 2014)*, 28-30 August, 2014, Bali, INDONESIA.
- Alireza Marefat, Rozita Aboki, Ali Jalooli, Erfan Shaghaghi, Mohammad Reza Jabbarpour and Rafidah Md Noor, "An adaptive overtaking maneuver assistant

system using VANET", *Asia Pacific Conference on Wireless and Mobile 2014 (APWiMob 2014)*, 28-30 August, 2014, Bali, INDONESIA.

- Mohammad Reza Jabbarpour, Ali Jalooli, Alireza Marefat and Rafidah Md Noor, "A Taxonomy-based Comparison of Vehicle Cloud Architectures", *The 3rd International Conference on Information and Computer Networks (ICICN 2015)*, 19-20 March 2015, Florence, Italy.
- Alireza Marefat, Mohammad Reza Jabbarpour and Rafidah Md Noor, "A Novel Comprehensive VANET-based Congestion Pricing Zone Management System", *The 14th International Conference on Traffic and Transportation Engineering (ICTTE 2015)*, February 24-25, 2015, Tehran, Iran. (Under review)

7.4. Future Work and New Directions

There is still a lot to do in the area of vehicle routing and traffic congestion mitigation. Further studies might focus on designing a smarter routing algorithm which considers different types of priorities for different types of vehicles such as taxi, bus, ambulances, police cars and fire trucks in its guidance procedure. Value of time (VOT) represents how much money the traveler is willing to pay to find and use optimal paths for time saving. By considering this preference as an input variable for AVCAF, we can contribute to the improvement of framework in our future work.

It would also be useful to develop a green extension of our proposed framework by considering an emission model in its routing mechanism. Hence, both vehicles' travel time and air pollutions can be reduced simultaneously. Also, AVCAF can be extended to perform in an unexplored environment (i.e. road map network) where the ant agents do not have an overall view from the search space. In this case, reinforcement learning algorithms or Markov decision processes can be used for optimal paths finding. Another interesting subject that should be studied is the investigation of self-adaptive parameter tunings. In this case, the stagnation measures can be useful to set algorithm's parameters such as α, β and ρ, autonomously.

On the theoretical side, other features of computing intelligence such as adaptation, flexibility and learning will be considered as an extension of AVCAF. Besides, we will analyze the convergence properties of our proposed approach. Convergence means whether a given algorithm is able to find an optimal solution when there are sufficient resources.

Reference

Abbas, Sagheer, Khan, M Saleem, Ahmed, Khalil, Abdullah, M, & Farooq, Umer. (2011). Bio-inspired neuro-fuzzy based dynamic route selection to avoid traffic congestion. *International Journal of Scientific and Engineering Research, 2*(6), 284-289.

Abbass, Hussein A. (2001). *MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach.* Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, *1*, (pp. 207-214).

Ahangarikiasari, Hojjatollah, Saraji, Mehdi Rahmani, & Torabi, Mohammad. (2013). Investigation of code complexity of an innovative algorithm based on ACO in weighted graph traversing and compare it to traditional ACO and Bellman-Ford. *Journal of Bioinformatics and Intelligent Control, 2*(1), 73-78.

Ahmed, Hazem, & Glasgow, Janice. (2012). Swarm intelligence: concepts, models and applications (pp. 1-51): Technical Report 2012-585, Queen's University, Kingston, Ontario, Canada K7L3N6.

Ahn, Chang Wook, & Ramakrishna, Rudrapatna S. (2002). A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation, 6*(6), 566-579.

Alhalabi, Sanaa M, Al-Qatawneh, Sokyna M, & Samawi, Venus W. (2008). *Developing a route navigation system using genetic algorithm.* 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2008), Damascus, (pp. 1-6).

Amadeo, Marica, Campolo, Claudia, Molinaro, Antonella, & Ruggeri, Giuseppe. (2009). *A WAVE-compliant MAC protocol to support vehicle-to-infrastructure non-safety applications.* IEEE International Conference on Communications Workshops, ICC Workshops 2009, Dresden, (pp. 1-6).

Ando, Yasushi, Fukazawa, Yoshiaki, Masutani, Osamu, Iwasaki, Hirotoshi, & Honiden, Shinichi. (2006). *Performance of pheromone model for predicting traffic congestion.* Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM New York, NY, USA, (pp. 73-80).

Ando, Yasushi, Masutani, Osamu, Sasaki, Hiroshi, Iwasaki, Hirotoshi, Fukazawa, Yoshiaki, & Honiden, Shinichi. (2006). Pheromone model: Application to traffic congestion prediction *Engineering Self-Organising Systems* (pp. 182-196): Springer.

André, M, & Hammarström, U. (2000). Driving speeds in Europe for pollutant emissions estimation. *Transportation Research Part D: Transport and Environment, 5*(5), 321-335.

Apostolopoulos, Theofanis, & Vlachos, Aristidis. (2010). Application of the firefly algorithm for solving the economic emissions load dispatch problem. *International Journal of Combinatorics, 2011*, 1-23. doi: 10.1155/2011/523806

AquaLab. (2002). Car-to-car cooperation for vehicular ad-hoc networks. Retrieved 18/6/2014, 2014, from http://aqualab.cs.northwestern.edu/projects/111-c3-car-to-car-cooperation-for-vehicular-ad-hoc-networks

Barr, R, Haas, ZJ, & Van Renesse, R. (2005). JiST/SWANS: java in simulation time/scalable wireless ad hoc network simulator. *http://jist.ece.cornell.edu*.

Barr, Rimon. (2004). Swans-scalable wireless ad hoc network simulator. *http://jist.ece.cornell.edu/docs.html*, 1-15.

Barth, Matthew, & Boriboonsomsin, Kanok. (2008). Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board, 2058*(1), 163-171.

Barth, Matthew, & Boriboonsomsin, Kanok. (2009). Traffic congestion and greenhouse gases. *ACCESS Magazine, 1*(35).

Batool, Fatima, & Khan, Shoad A. (2005). *Traffic estimation and real time prediction using adhoc networks.* Proceedings of the IEEE Symposium on Emerging Technologies, Islamabad, (pp. 264-269).

Bedi, Punam, Mediratta, Neha, Dhand, Silky, Sharma, Ravish, & Singhal, Archana. (2007). *Avoiding traffic jam using ant colony optimization-a novel approach.* International Conference on Conference on Computational Intelligence and Multimedia Applications, Sivakasi, Tamil Nadu, (pp. 61-67).

Benedettini, Stefano, Roli, Andrea, & Di Gaspero, Luca. (2008). Two-level ACO for haplotype inference under pure parsimony *Ant Colony Optimization and Swarm Intelligence* (pp. 179-190): Springer.

Bennett, Jonathan. (2010). *OpenStreetMap*: Packt Publishing Ltd.

Birkin, Mark, Turner, Andy, & Wu, Belinda. (2006). *A synthetic demographic model of the UK population: Methods, progress and problems.* 36th Annual Conference Regional Science Association International British and Irish Section, (pp. 1-6)

Blum, Christian. (2005). Ant colony optimization: introduction and recent trends. *Physics of Life reviews, 2*(4), 353-373.

Blum, Christian, & Blesa, Maria J. (2005). New metaheuristic approaches for the edge-weighted *k*-cardinality tree problem. *Computers & Operations Research, 32*(6), 1355-1377.

Blum, Christian, & Dorigo, Marco. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34*(2), 1161-1172.

Blum, Christian, & Li, Xiaodong. (2008). Swarm intelligence in optimization. *Swarm Intelligence*: Springer Berlin Heidelberg, (pp. 43-85).

Blum, Christian, & Merkle, Daniel. (2008). *Swarm intelligence: introduction and applications*: Springer.

Blum, Christian, Vallès, Mateu Yábar, & Blesa, Maria J. (2008). An ant colony optimization algorithm for DNA sequencing by hybridization. *Computers & Operations Research*, *35*(11), 3620-3635.

Bohm, A, Lidstrom, K, Jonsson, Magnus, & Larsson, Tony. (2010). *Evaluating CALM M5-based vehicle-to-vehicle communication in various road settings through field trials*. 35th IEEE Conference on Local Computer Networks (LCN), Denver, CO, (pp. 613-620).

Bonabeau, Eric, Dorigo, Marco, & Theraulaz, Guy. (1999). *Swarm intelligence: from natural to artificial systems*: Oxford university press.

Boryczka, Mariusz, & Bura, Wojciech. (2013). Ant colony optimization for the multi-criteria vehicle navigation problem. *Agent-Based Optimization* (pp. 29-53): Springer Berlin Heidelberg.

Brucker, Peter, & Brucker, P. (2007). *Scheduling algorithms* (Vol. 3, pp. 1-367): Springer.

Bullnheimer, Bernd, Hartl, Richard F, & Strauss, Christine. (1997). A new rank based version of the Ant System. A computational study. *Working paper series*, 1-16.

Bura, Wojciech, & Boryczka, Mariusz. (2010). Ant colony system in ambulance navigation. *Journal of Medical Informatics & Technologies, 15*, 115-124.

Bura, Wojciech, & Boryczka, Mariusz. (2012). Ant colony optimization for the pareto front approximation in vehicle navigation. *Computational Collective Intelligence. Technologies and Applications* (pp. 493-502): Springer.

Busanelli, Stefano, Ferrari, Gianluigi, & Giorgio, Vito Andrea. (2011). I2V highway and urban vehicular networks: a comparative analysis of the impact of mobility on broadcast data dissemination. *Journal of Communications, 6*(1).

Busanelli, Stefano, Ferrari, Gianluigi, & Panichpapiboon, Sooksan. (2009). *Efficient broadcasting in IEEE 802.11 networks through irresponsible forwarding*. IEEE Global Telecommunications Conference, GLOBECOM 2009, Honolulu, HI, (pp. 1-6).

Cagara, Daniel, Bazzan, Ana LC, & Scheuermann, Björn. (2014). *Getting you faster to work: a genetic algorithm approach to the traffic assignment problem*. Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion, (pp. 105-106).

Casillas, J, Cordón, O, & Herrera, F. (2000). Learning cooperative fuzzy linguistic rules using ant colony algorithms: Technical report DECSAI-00-01-19, Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain.

Chakraborty, Basabi. (2004). Ga-based multiple route selection for car navigation *Applied Computing* (pp. 76-83): Springer.

Chakraborty, Basabi. (2005). Simultaneous multiobjective multiple route selection using genetic algorithm for car navigation. *Pattern Recognition and Machine Intelligence* (pp. 696-701): Springer.

Chakraborty, Basabi, & Chen, Rung Ching. (2009). *Fuzzy-genetic approach for incorporation of driver's requirement for route selection in a car navigation system.* IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2009, Jeju Island, (pp. 1645 - 1649).

Chakraborty, Basabi, Maeda, Takeaki, & Chakraborty, Goutam. (2005). *Multiobjective route selection for car navigation system using genetic algorithm.* Proceedings of the 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, SMCia/05, (pp. 190-195).

Chan, Kit Yan, Dillon, Tharam S, & Chang, Elizabeth. (2013). An intelligent particle swarm optimization for short-term traffic flow forecasting using on-road sensor systems. *Industrial Electronics, IEEE Transactions on, 60*(10), 4714-4725.

Chang, Ben-Jye, Huang, Bo-Jhang, & Liang, Ying-Hsin. (2008). *Wireless sensor network-based adaptive vehicle navigation in multihop-relay WiMAX networks.* 22nd International Conference on Advanced Information Networking and Applications, AINA 2008, Okinawa, (pp. 56-63).

Chang, Xinjie. (1999). *Network simulations with OPNET.* Proceedings of the 31st conference on Winter simulation: Simulation: a bridge to the future, (pp. 307-314).

Chen, Qi, Schmidt-Eisenlohr, Felix, Jiang, Daniel, Torrent-Moreno, Marc, Delgrossi, Luca, & Hartenstein, Hannes. (2007). *Overhaul of IEEE 802.11 modeling and simulation in ns-2.* Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, (pp. 159-168).

Chen, Xin, Yang, Zhao-Sheng, & Wang, Hai-Yang. (2006). *A multi-agent urban traffic control system cooperated with dynamic route guidance.* International Conference on Machine Learning and Cybernetics, Dalian, China, (pp. 330 - 335).

Cheung, Sing Yiu, Coleri, Sinem, Dundar, Baris, Ganesh, Sumitra, Tan, Chin-Woo, & Varaiya, Pravin. (2005). Traffic measurement and vehicle classification with single magnetic sensor. *Transportation research record: journal of the transportation research board, 1917*(1), 173-181.

Chien, Steven I-Jy, & Kuchipudi, Chandra Mouly. (2003). Dynamic travel time prediction with real-time and historic data. *Journal of transportation engineering, 129*(6), 608-616.

Chiu, Yi-Chang, Bottom, Jon, Mahut, Michael, Paz, Alex, Balakrishna, Ramachandran, Waller, Travis, & Hicks, Jim. (2011). Dynamic traffic assignment: A primer. *Transportation Research E-Circular*(E-C153).

Choffnes, David R, & Bustamante, Fabián E. (2005). *An integrated mobility and traffic model for vehicular wireless networks.* Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, (pp. 69 - 78).

Claes, Rutger, & Holvoet, Tom. (2011). *Ant colony optimization applied to route planning using link travel time predictions.* IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), Shanghai, (pp. 358 - 365).

Claes, Rutger, & Holvoet, Tom. (2012). Cooperative ant colony optimization in traffic route calculations. *Advances on Practical Applications of Agents and Multi-Agent Systems* (pp. 23-34): Springer.

Coifman, Benjamin A, & Mallika, Ramachandran. (2007). Distributed surveillance on freeways emphasizing incident detection and verification. *Transportation research part A: policy and practice, 41*(8), 750-767.

Cong, Zhe, De Schutter, Bart, & Babuska, Robert. (2011). *A new ant colony routing approach with a trade-off between system and user optimum.* 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, (pp. 1369 - 1374).

Cong, Zhe, De Schutter, Bart, & Babuška, Robert. (2013). Ant Colony Routing algorithm for freeway networks. *Transportation Research Part C: Emerging Technologies, 37*, 1-19.

Costa, Daniele, & Hertz, Alain. (1997). Ants can colour graphs. *Journal of the operational Research Society, 48*(3), 295-305.

Daganzo, Carlos F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological, 28*(4), 269-287.

Darus, Mohamad Yusof, & Bakar, Kamalrulnizam Abu. (2011). A review of congestion control algorithm for event-driven safety messages in vehicular networks. *channels, 8*(5), 49-53.

Davies, Cedric, & Lingras, Pawan. (2003). Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks. *European Journal of Operational Research, 144*(1), 27-38.

De Campos, Luis M, Fernandez-Luna, Juan M, Gámez, José A, & Puerta, José M. (2002). Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning, 31*(3), 291-311.

Delavar, MR, Samadzadegan, F, & Pahlavani, P. (2004). A GIS-assisted optimal urban route finding approach based on genetic algorithms. *International archives of photogrammetry remote sensing and spatial information sciences, 35*(Part 2), 305-308.

Deneubourg, J-L, Aron, Serge, Goss, Simon, & Pasteels, Jacques Marie. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior, 3*(2), 159-168.

Deng, Yanfang, Tong, Hengqing, & Zhang, Xiedong. (2010). *Dynamic shortest path in stochastic traffic networks based on fluid neural network and particle swarm optimization.* Sixth International Conference on Natural Computation (ICNC), Yantai, Shandong, (pp. 2325 - 2329).

Desai, Prajakta, Loke, Seng Wai, Desai, Aniruddha, & Singh, Jugdutt. (2013). CARAVAN: congestion avoidance and route allocation using virtual agent negotiation. *IEEE Transactions on Intelligent Transportation Systems, 14*(3), 1197-1207.

Dezani, Henrique, Bassi, Regiane DS, Marranghello, Norian, Gomes, Luís, Damiani, Furio, & Nunes da Silva, Ivan. (2014). Optimizing urban traffic flow using Genetic Algorithm with Petri net analysis as fitness function. *Neurocomputing, 124*, 162-167.

Dhillon, Santpal Singh, & Van Mieghem, Piet. (2007). Performance analysis of the AntNet algorithm. *Computer Networks, 51*(8), 2104-2125.

Dhillon, SS, & Van Mieghem, Piet. (2007). Performance analysis of the AntNet algorithm. *Computer Networks, 51*(8), 2104-2125.

Di Caro, Gianni. (2004). *Ant colony optimization and its application to adaptive routing in telecommunication networks* (PhD thesis, Faculté des Sciences Appliquées, Université Libre de Bruxelles, Brussels, Belgium).

Di Caro, Gianni, & Dorigo, Marco. (2011). AntNet: distributed stigmergetic control for communications networks *arXiv preprint arXiv:1105.5449*.

Dia, Hussein. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research, 131*(2), 253-261.

Dietzel, Stefan, Petit, Jonathan, Kargl, Frank, & Scheuermann, Björn. (2014). In-Network Aggregation for Vehicular Ad Hoc Networks. *IEEE Communication Surveys & Tutorials, 16*(4), 1909-1932.

Dijkstra, Edsger W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik, 1*(1), 269-271.

Dimitrakopoulos, George, & Demestichas, Panagiotis. (2010). Intelligent transportation systems. *IEEE Vehicular Technology Magazine, 5*(1), 77-84.

Ding, J-W, Wang, C-F, Meng, F-H, & Wu, T-Y. (2010). Real-time vehicle route guidance using vehicle-to-vehicle communication. *IET communications, 4*(7), 870-883.

Doolan, Ronan, & Muntean, Gabriel-Miro. (2014). *Time-Ants: an innovative temporal and spatial ant-based vehicular Routing Mechanism.* Paper presented at the IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI.

Dorigo, M, & Stützle, T. (2004). AntNet: an ACO Algorithm for Data Network Routing *Ant Colony Optimization* (1 ed., pp. 223 - 260): MIT Press.

Dorigo, Marco. (1992). *Optimization, learning and natural algorithms* (Ph. D. Thesis, Politecnico di Milano, Italy).

Dorigo, Marco, & Birattari, Mauro. (2010). Ant colony optimization *Encyclopedia of Machine Learning* (pp. 36-39): Springer.

Dorigo, Marco, Caro, Gianni, & Gambardella, Luca. (1999). Ant algorithms for discrete optimization. *Artificial life, 5*(2), 137-172.

Dorigo, Marco, & Gambardella, Luca Maria. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53-66.

Dorigo, Marco, Maniezzo, Vittorio, & Colorni, Alberto. (1991). The ant system: an autocatalytic optimizing process. Dipartimento di Elettronica, Politecnico di Milano, Italy: Technical Report 91-016.

Dorigo, Marco, Maniezzo, Vittorio, & Colorni, Alberto. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26*(1), 29-41.

Dorigo, Marco, & Stützle, Thomas. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. *Handbook of metaheuristics* (pp. 250-285): Springer.

Dorigo, Marco, & Stützle, Thomas. (2010). Ant colony optimization: overview and recent advances. *Handbook of metaheuristics* (pp. 227-263): Springer.

Elbeltagi, Emad, Hegazy, Tarek, & Grierson, Donald. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced engineering informatics, 19*(1), 43-53.

En, De, Wei, Huanghe, Yang, Jinxian, Wei, Nana, Chen, Xiaoguang, & Liu, Yangyang. (2012). Analysis of the shortest path of GPS vehicle navigation system based on genetic algorithm. *Electrical, Information Engineering and Mechatronics* (pp. 413-418): Springer.

Estrin, Deborah, Handley, Mark, Heidemann, John, McCanne, Steven, Xu, Ya, & Yu, Haobo. (2000). Network visualization with nam, the vint network animator. *Computer, 33*(11), 63-68.

Eugster, Patrick Th, Felber, Pascal A, Guerraoui, Rachid, & Kermarrec, Anne-Marie. (2003). The many faces of publish/subscribe. *ACM Computing Surveys (CSUR), 35*(2), 114-131.

Fall, Kevin, & Varadhan, Kannan. (2014). The network simulator (ns-2). Retrieved 12/12/2014, from http://www.isi.edu/nsnam/ns

Fan, Hui, Hua, Zhen, Li, Jin-Jiang, & Yuan, Da. (2004). *Solving a shortest path problem by ant algorithm.* Proceedings of 2004 International Conference on Machine Learning and Cybernetics, *5*, (pp. 3174 - 3177).

Finch, Tony. (2009). Incremental calculation of weighted mean and variance. University of Cambridge: Computing Service.

Flood, Merrill M. (1956). The traveling-salesman problem. *Operations Research, 4*(1), 61-75.

Ford, LR, & Fulkerson, Delbert Ray. (1962). *Flows in networks* (Vol. 1962): Princeton Princeton University Press.

Foroughi, R, Montazer, Gh A, & Sabzevari, R. (2008). Design of a new urban traffic control system using modified ant colony optimization approac. *Iranian Journal of Science and Technology, 32*(B2), 167.

Fu, Liping. (2001). An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B: Methodological, 35*(8), 749-765.

Fuellerer, Guenther, Doerner, Karl F, Hartl, Richard F, & Iori, Manuel. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research, 36*(3), 655-673.

Gambardella, Luca Maria, & Dorigo, Marco. (1995). *Ant-Q: a reinforcement learning approach to the traveling salesman problem.* International Conference on Mashine Learning (ICML), Tahoe City, California, (pp. 252-260).

Gambardella, Luca Maria, & Dorigo, Marco. (2000). An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing, 12*(3), 237-255.

Gandomi, Amir Hossein, Yang, Xin-She, & Alavi, Amir Hossein. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*(1), 17-35.

García-Gonzalo, E, & Fernández-Martínez, JL. (2012). A brief historical review of particle swarm optimization (PSO). *Journal of Bioinformatics and Intelligent Control, 1*(1), 3-16.

García-Martínez, Carlos, Cordón, Oscar, & Herrera, Francisco. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research, 180*(1), 116-148.

Ge, Yan, Wang, Jian, Wang, Guijia, & Jiang, Feng. (2011). Urban vehicle routing research based on ant colony algorithm and traffic rule restriction *High Performance Networking, Computing, and Communication Systems* (pp. 95-102): Springer.

Gen, Mitsuo, Cheng, Runwei, & Wang, Qing. (1997). *Genetic algorithms for solving shortest path problems*. IEEE International Conference on Evolutionary Computation, Indianapolis, IN, USA, (pp. 401-406).

Ghazy, Ayman M, EL-Licy, Fatma, & Hefny, Hesham A. (2012). Threshold based AntNet algorithm for dynamic traffic routing of road networks. *Egyptian Informatics Journal, 13*(2), 111-121.

Ghosal, Prasun, Chakraborty, Arijit, & Banerjee, Sabyasachee. (2013). *Honey bee based vehicular traffic optimization and management.* Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), (Vol. 2, pp. 455-463).

Ghoseiri, Keivan, & Nadjari, Behnam. (2010). An ant colony optimization algorithm for the bi-objective shortest path problem. *Applied Soft Computing, 10*(4), 1237-1246.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*: Addison-Wesley.

Golden, Bruce L, Raghavan, Subramanian, & Wasil, Edward A. (2008). *The vehicle routing problem: latest advances and new challenges* (Vol. 43): Springer.

Gong, Qiuming, Li, Yaoyu, & Peng, Zhong-Ren. (2008). Trip-based optimal power management of plug-in hybrid electric vehicles. *IEEE Transactions on Vehicular Technology, 57*(6), 3393-3401.

Gradinescu, Victor, Gorgorin, Cristian, Diaconescu, Raluca, Cristea, Valentin, & Iftode, Liviu. (2007). *Adaptive traffic lights using car-to-car communication.* IEEE 65th Vehicular Technology Conference, VTC2007-Spring, Dublin, (pp. 21-25).

Hallam, Nasreddine, Hartley, Michael, Blanchfield, Peter, & Kendall, Graham. (2004). *Optimisation in a road traffic system using collaborative search.* IEEE International Conference on Systems, Man and Cybernetics, (Vol. 2, pp. 2008 - 2012).

Hansen, Pierre. (1980). Bicriterion path problems *Multiple criteria decision making theory and application* (pp. 109-127): Springer.

Härri, Jérôme, Filali, Fethi, Bonnet, Christian, & Fiore, Marco. (2006). *VanetMobiSim: generating realistic mobility patterns for VANETs*. Proceedings of the 3rd international workshop on Vehicular ad hoc networks, (pp. 96-97).

Harrison, David. (1989). Xgraph version 11.3. 2. *University of California Berkeley Electronics Research Laboratory*.

Hart, Peter E, Nilsson, Nils J, & Raphael, Bertram. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*(2), 100-107.

Hartenstein, Hannes, & Laberteaux, Kenneth P. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine, 46*(6), 164-171.

He, Zongjian, Cao, Jiannong, & Li, Tao. (2012). *MICE: a real-time traffic estimation based vehicular path planning solution using VANETs.* International Conference on Connected Vehicles and Expo (ICCVE), Beijing, (pp. 172-178).

Hodge, Victoria, Krishnan, Rajesh, Jackson, Tom, Austin, Jim, & Polak, John. (2011). *Short-term traffic prediction using a binary neural network.* 43rd Annual UTSG Conference, Open University, Milton Keynes, UK, (pp. 1-12).

Hogie, Luc, Bouvry, Pascal, & Guinand, Frédéric. (2006). An overview of manets simulation. *Electronic notes in theoretical computer science, 150*(1), 81-101.

Horoba, Christian. (2010). Exploring the runtime of an evolutionary algorithm for the multi-objective shortest path problem. *Evolutionary computation, 18*(3), 357-381.

Hrizi, Fatma, & Filali, Fethi. (2010). *simITS: an integrated and realistic simulation platform for vehicular networks.* Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, (pp. 32-36).

Hu, Jianming, Gao, Pan, Yao, Yunfei, & Xie, Xudong. (2014). *Traffic flow forecasting with particle swarm optimization and support vector regression.* 17th IEEE International Conference on Intelligent Transportation Systems (ITSC), Qingdao, (pp. 2267 - 2268).

Hu, L, Gu, ZQ, Huang, J, Yang, Y, & Song, X. (2008). Research and realization of optimum route planning in vehicle navigation systems based on a hybrid genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 222*(5), 757-763.

Inagaki, Jun, Haseyama, M, & Kitajima, H. (1999). A method of determining various solutions for routing application with a genetic algorithm. *Trans. Inst. Electron. Inf. Commun. Eng. DI (Japan), J82D-I (8)*, 1102-1111.

Inagaki, Jun, Haseyama, Miki, & Kitajima, Hideo. (1999). *A genetic algorithm for determining multiple routes and its applications.* Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, ISCAS'99, Orlando, FL, (Vol. 6, pp. 137 - 140).

Issariyakul, Teerawat, & Hossain, Ekram. (2011). *Introduction to network simulator NS2* (pp. 1-512): Springer.

Jerbi, Moez, Senouci, S-M, Rasheed, Tinku, & Ghamri-Doudane, Yacine. (2007). *An infrastructure-free traffic information system for vehicular networks.* IEEE 66th Vehicular Technology Conference, VTC-2007-Fall, Baltimore, MD, (pp. 2086 - 2090).

Jiang, Daniel, & Delgrossi, Luca. (2008). *IEEE 802.11 p: towards an international standard for wireless access in vehicular environments.* IEEE Vehicular Technology Conference, VTC-Spring, Singapore, (pp. 2036 - 2040).

Jiang, Yong, Wang, Wan-Liang, & Zhao, Yan-Wei. (2007). *Solving the shortest path problem in vehicle navigation system by ant colony algorithm.* Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, (pp. 188-192).

Jin, FH, Hong, BR, & Gao, QJ. (2002). Path planning for free-flying space robot using ant algorithm. *Robot, 24*(6), 526-529.

Kammoun, Habib M, Kallel, Ilhem, Alimi, Adel M, & Casillas, Jorge. (2010). *An adaptive vehicle guidance system instigated from ant colony behavior.* IEEE International Conference on Systems Man and Cybernetics (SMC), Istanbul, (pp. 2948 - 2955).

Kammoun, Habib M, Kallel, Ilhem, Alimi, Adel M, & Casillas, Jorge. (2011). *Improvement of the road traffic management by an ant-hierarchical fuzzy system.* IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS), Paris, (pp. 38 - 45).

Kammoun, Habib M, Kallel, Ilhem, Casillas, Jorge, Abraham, Ajith, & Alimi, Adel M. (2014). Adapt-Traf: an adaptive multiagent road traffic management system based on hybrid ant-hierarchical fuzzy model. *Transportation Research Part C: Emerging Technologies, 42*, 147-167.

Kanoh, Hitoshi. (2007). Dynamic route planning for car navigation systems using virus genetic algorithms. *International Journal of Knowledge-based and Intelligent Engineering Systems, 11*(1), 65-78.

Kanoh, Hitoshi, & Hara, Kenta. (2008). *Hybrid genetic algorithm for dynamic multi-objective route planning with predicted traffic in a real-world road network.* Proceedings of the 10th annual conference on Genetic and evolutionary computation, (pp. 657-664).

Kanoh, Hitoshi, & Nakamura, Tomohiro. (2000). *Knowledge based genetic algorithm for dynamic route selection.* Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Brighton, (Vol. 2, pp. 616-619).

Karaboga, Dervis. (2005). An idea based on honey bee swarm for numerical optimization: Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Karaboga, Dervis, & Akay, Bahriye. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*(1), 108-132.

Karaboga, Dervis, & Basturk, Bahriye. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization, 39*(3), 459-471.

Karaboga, Dervis, & Basturk, Bahriye. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing, 8*(1), 687-697.

Karaboga, Dervis, Gorkemli, Beyza, Ozturk, Celal, & Karaboga, Nurhan. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review, 42*(1), 21-57.

Karagiannis, Georgios, Altintas, Onur, Ekici, Eylem, Heijenk, Geert, Jarupan, Boangoat, Lin, Kenneth, & Weil, Timothy. (2011). Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials, 13*(4), 584-616.

Karbassi, Abdolreza, & Barth, Matthew. (2003). *Vehicle route prediction and time of arrival estimation techniques for improved transportation system management.* IEEE Intelligent Vehicles Symposium, (pp. 511-516).

Karimi, A, Hegyi, A, De Schutter, B, Hellendoorn, J, & Middelham, F. (2004). *Integrated model predictive control of dynamic route guidance information systems and ramp metering.* 7th International IEEE Conference on Intelligent Transportation Systems, (pp. 491-496).

Karnadi, Feliz K, Mo, Zhi Hai, & Lan, Kun-chan. (2007). *Rapid generation of realistic mobility models for VANET.* IEEE Wireless Communications and Networking Conference, WCNC, Kowloon, (pp. 2506-2511).

Kennedy, J, & Eberhart, RC. (1995). *Particle swarm optimisation.* Proceedings IEEE International Conference on Neural Networks, Piscataway, NJ, (pp. 1-6).

Kennedy, James, Kennedy, James F, & Eberhart, Russell C. (2001). *Swarm intelligence*: Morgan Kaufmann.

Khanjary, Mohammad, & Hashemi, Seyyed Mohsen. (2012). *Route guidance systems: review and classification.* Proceedings of the 6th Euro American Conference on Telematics and Information Systems, (pp. 269-275).

Kim, Byung-Ki, Jo, Jung-Bok, Kim, Jong-Ryul, & Gen, Mitsuo. (2009). Optimal route search in car navigation systems by multi-objective genetic algorithms. *International Journal of Information Systems for Logistics and Management, 4*(2), 9-18.

Köhler, Ekkehard, Langkau, Katharina, & Skutella, Martin. (2002). Time-expanded graphs for flow-dependent transit times *Algorithms—ESA 2002* (pp. 599-611): Springer.

Konishi, Kazuki, Maeda, Kumiko, Sato, Kazuki, Yamasaki, Akiko, Yamaguchi, Hirozumi, Yasumoto, Keiichi, & Higashinoz, T. (2005). *MobiREAL: simulator-evaluating manet applications in real environments.* 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, (pp. 499-502).

Korb, Oliver, Stützle, Thomas, & Exner, Thomas E. (2007). An ant colony optimization approach to flexible protein–ligand docking. *Swarm Intelligence, 1*(2), 115-134.

Kotsialos, Apostolos, Papageorgiou, Markos, Mangeas, Morgan, & Haj-Salem, Habib. (2002). Coordinated and integrated control of motorway networks via non-linear optimal control. *Transportation Research Part C: Emerging Technologies, 10*(1), 65-84.

Kponyo, JJ, Kuang, Yujun, & Li, Zejiao. (2012). *Real time status collection and dynamic vehicular traffic control using ant colony optimization.* International Conference on Computational Problem-Solving (ICCP), Leshan, (pp. 69-72).

Krajzewicz, Daniel, Erdmann, Jakob, Behrisch, Michael, & Bieker, Laura. (2012). Recent development and applications of SUMO–simulation of urban mobility. *International Journal On Advances in Systems and Measurements, 5*(3 and 4), 128-138.

Krishnanand, KR, Nayak, Santanu Kumar, Panigrahi, KB, & Rout, PK. (2009). *Comparative study of five bio-inspired evolutionary optimization techniques.* World Congress on Nature & Biologically Inspired Computing, NaBIC, Coimbatore, (pp. 1231-1236).

Król, Dariusz, & Mrożek, Maciej. (2011). Swarm-based multi-agent simulation: a case study of urban traffic flow in the city of wroclaw *Computational Collective Intelligence. Technologies and Applications* (pp. 191-200): Springer.

Kroon, Ronald. (2002). *Dynamic vehicle routing using ant based control* (Master theis, Delft University of Technology, Delft, Nederland).

Kroon, Ronald, & Rothkrantz, Leon. (2003). *Dynamic vehicle routing using an ABC-algorithm*. Intelligent Routing and Travel Information Systems, TRAIL Research School, Delft, Nederland, (pp. 21-28).

Kurihara, Satoshi. (2013). Traffic-congestion forecasting algorithm based on pheromone communication model. *Ant Colony Optimization-Techniques and Applications*, 1-13.

Kurihara, Satoshi, Tamaki, Hiroshi, Numao, Masayuki, Yano, Junji, Kagawa, Kouji, & Morita, Tetsuo. (2009). *Traffic congestion forecasting based on pheromone communication model for intelligent transport systems.* Proceedings of the Eleventh conference on Congress on Evolutionary Computation, (pp. 2879-2884).

Kwon, Jaimyoung, & Petty, Karl. (2005). Travel time prediction algorithm scalable to freeway networks with many nodes with arbitrary travel routes. *Transportation Research Record: Journal of the Transportation Research Board, 1935*(1), 147-153.

Lalwani, Soniya, Singhal, Sorabh, Kumar, Rajesh, & Gupta, Nilama. (2013). A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm. *Transactions on Combinatorics*, *2*(1), 39-101.

LaValle, Steven M. (2006). *Planning algorithms*: Cambridge university press.

Lee, Jungkyu, & Yang, Jihoon. (2012). A fast and scalable re-routing algorithm based on shortest path and genetic Algorithms. *International Journal of Computers Communications & Control (IJCCC), 7*(3), 482-493.

Leguizamon, Guillermo, & Michalewicz, Zbigniew. (1999). *A new version of ant system for subset problems.* Proceedings of the 1999 Congress on Evolutionary Computation, (pp. 1-7)

Lessing, Lucas, Dumitrescu, Irina, & Stützle, Thomas. (2004). A comparison between ACO algorithms for the set covering problem *Ant Colony Optimization and Swarm Intelligence* (pp. 1-12): Springer.

Leung, Yee, Li, Guo, & Xu, Zong-Ben. (1998). A genetic algorithm for the multiple destination routing problems. *IEEE Transactions on Evolutionary Computation, 2*(4), 150-161.

Lima, Richardson. (2009). AntNet algorithm implementation on NS-2.33. *Computer Science Center of the Federal University of Pernambuco at Recife, Brazil*.

Liu, Chi Harold, Fan, Jun, Branch, Joel W, & Leung, Kin K. (2014). Towards QoI and energy-efficiency in Internet-of-Things sensory environments. *IEEE Transactions on Emerging Topics in Computing*(99), 473-487. doi: 10.1109/TETC.2014.2364915

Liu, Jingyu, Fang, Yanjun, & Liu, Yijian. (2007). *Ant colony system algorithm for path routing of urban traffic vehicles.* IEEE International Conference on Automation and Logistics, Jinan, (pp. 1902-1907).

Mak, Tony K, Laberteaux, Kenneth P, Sengupta, Raja, & Ergen, Mustafa. (2009). Multichannel medium access control for dedicated short-range communications. *IEEE Transactions on Vehicular Technology, 58*(1), 349-366.

Mangharam, Rahul, Weller, Daniel, Rajkumar, Raj, Mudalige, Priyantha, & Bai, Fan. (2006). *Groovenet: a hybrid simulator for vehicle-to-vehicle networks.* Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, San Jose, CA, (pp. 1-8).

Maniezzo, Vittorio. (1999). Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS journal on computing, 11*(4), 358-369.

Maniezzo, Vittorio, & Carbonaro, Antonella. (2000). An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems, 16*(8), 927-935.

Martin, J. (2001). GloMoSim. global mobile information systems simulation library. UCLA Parallel Computing Laboratory.

Martinez, Francisco J, Cano, J-C, Calafate, Carlos T, & Manzoni, Pietro. (2008). *Citymob: a mobility model pattern generator for VANETs.* IEEE International Conference on Communications Workshops, ICC Workshops' 08, Beijing, (pp. 370-374).

Martinez, Francisco J, Fogue, Manuel, Toh, Chai-Keong, Cano, Juan-Carlos, Calafate, Carlos T, & Manzoni, Pietro. (2013). Computer simulations of VANETs using realistic city topologies. *Wireless personal communications, 69*(2), 639-663.

Martinez, Francisco J, Toh, Chai-Keong, Cano, Juan-Carlos, Calafate, Carlos T, & Manzoni, Pietro. (2011a). A street broadcast reduction scheme (SBR) to mitigate the broadcast storm problem in VANETs. *Wireless personal communications, 56*(3), 559-572.

Martinez, Francisco J, Toh, Chai Keong, Cano, Juan-Carlos, Calafate, Carlos T, & Manzoni, Pietro. (2011b). A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless Communications and Mobile Computing, 11*(7), 813-828.

Martins, Ernesto Queiros Vieira. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research, 16*(2), 236-245.

Maslekar, Nitin, Boussedjra, Mounir, Mouzna, Joseph, & Labiod, Houda. (2011). *VANET based adaptive traffic signal control.* 73rd IEEE Vehicular Technology Conference, VTC-Spring, Budapest, (pp. 1-5).

McCanne, Steven, Floyd, Sally, Fall, Kevin, & Varadhan, Kannan. (1997). Network simulator ns-2.

Merkle, Daniel, & Middendorf, Martin. (2003). Ant colony optimization with global pheromone evaluation for scheduling a single machine. *Applied Intelligence, 18*(1), 105-111.

Merkle, Daniel, Middendorf, Martin, & Schmeck, Hartmut. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation, 6*(4), 333-346.

Miller, Jeffrey, & Horowitz, Ellis. (2007). *Freesim-a free real-time freeway traffic simulator.* IEEE Intelligent Transportation Systems Conference (ITSC), Seattle, WA, (pp. 18-23).

Mohemmed, Ammar W, Sahoo, Nirod Chandra, & Geok, Tan Kim. (2008). Solving shortest path problem using particle swarm optimization. *Applied Soft Computing, 8*(4), 1643-1653.

Mori, Hiroko, Kitaoka, Hironobu, & Teramoto, Eiji. (2006). Traffic simulation for predicting traffic situations at expo 2005. *R&D Review of Toyota CRDL, 41*(4), 45-51.

Mu'azu, Abubakar Aminu, Lawal, Ibrahim A, Haruna, Lawal, & Rabiu, Shamsuddeen. (2013). A comparative study for the choice of vehicular ad hoc networks (VANETs) simulator. *Journal of Applied Sciences & Environmental Sustainability, 1*(2), 33-51.

Multi-Agent Transport Simulation (MATSim) hompage. (2008). Retrieved 4/7/2014, from http://www.matsim.org/

Nahar, Shahrizul Anuar Abu, & Hashim, Fazida Hanim. (2011). *Modelling and analysis of an efficient traffic network using ant colony optimization algorithm.* Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), Bali, (pp. 32-36).

Nakrani, Sunil, & Tovey, Craig. (2004). On honey bees and dynamic server allocation in internet hosting centers. *Adaptive Behavior, 12*(3-4), 223-240.

Nanayakkara, Suranga Chandima, Srinivasan, Dipti, Lup, Lai Wei, German, Xavier, Taylor, Elizabeth, & Ong, SH. (2007). *Genetic algorithm based route planner for large urban street networks.* IEEE Congress on Evolutionary Computation (CEC), Singapore, (pp. 4469-4474).

Nanthawichit, Chumchoke, Nakatsuji, Takashi, & Suzuki, Hironori. (2003). Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. *Transportation Research Record: Journal of the Transportation Research Board, 1855*(1), 49-59.

Narayanam, Ramasuri, & Narahari, Yadati. (2010). A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*(99), 1-18.

Narzt, W, Wilflingseder, U, Pomberger, G, Kolb, D, & Hortner, H. (2010). Self-organising congestion evasion strategies using ant-based pheromones. *IET Intelligent Transport Systems, 4*(1), 93-102.

Nazari, Sara, Meybodi, M Reza, Salehigh, M Ali, & Taghipour, Sara. (2008). *An advanced algorithm for finding shortest path in car navigation system.* International Conference on Intelligent Networks and Intelligent Systems (ICINIS'08), Wuhan, (pp. 671-674).

The Network Simulator Wiki. (2014, 4/11/2011). Retrieved 29/6/2014, from http://nsnam.isi.edu/nsnam/index.php/Main_Page

Neuman, B Clifford. (1994). *Readings in distributed computing systems*: IEEE Computer Society Press Los Alamitos, CA, USA.

Ng, SC, Cheung, CC, Leung, SH, & Luk, A. (2003). *Fast convergence for backpropagation network with magnified gradient function.* Proceedings of the International Joint Conference on Neural Networks, (Vol. 3, pp. 1903-1908).

Noto, Masato, & Sato, Hiroaki. (2000). *A method for the shortest path search by extended Dijkstra algorithm.* IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, (Vol. 3, pp. 2316-2320).

Ok, Seung-Ho, Seo, Woo-Jin, Ahn, Jin-Ho, Kang, Sungho, & Moon, Byungin. (2011). An ant colony optimization approach for the preference-based shortest path search. *Journal of the Chinese Institute of Engineers, 34*(2), 181-196.

Ong, HC, Mahlia, TMI, & Masjuki, HH. (2011). A review on emissions and mitigation strategies for road transport in Malaysia. *Renewable and Sustainable Energy Reviews, 15*(8), 3516-3522.

Otero, Fernando EB, Freitas, Alex A, & Johnson, Colin G. (2008). cAnt-Miner: an ant colony classification algorithm to cope with continuous attributes *Ant colony optimization and swarm intelligence* (pp. 48-59): Springer.

Pan, J, Khan, Mohammad A, Sandu Popa, Iulian, Zeitouni, Karine, & Borcea, Cristian. (2012). *Proactive vehicle re-routing strategies for congestion avoidance.* 8th International Conference on Distributed Computing in Sensor Systems (DCOSS), Hangzhou, (pp. 265-272).

Panigrahi, Bijaya Ketan, Hiot, Lim Meng, & Shi, Yuhui. (2010). *Handbook of swarm intelligence: concepts, principles and applications* (Vol. 8): Springer.

Panigrahi, Bijaya Ketan, Shi, Yuhui, & Lim, Meng-Hiot. (2011). *Handbook of swarm intelligence*: Springer.

Papaleondiou, Leontios G, & Dikaiakos, Marios D. (2009). *Trafficmodeler: a graphical tool for programming microscopic traffic simulators through high-level abstractions.* 69th IEEE Vehicular Technology Conference, VTC-Spring, Barcelona, (pp. 1-5).

Park, Jungme, Li, Dai, Murphey, Yi Lu, Kristinsson, Johannes, McGee, Ryan, Kuang, Ming, & Phillips, Tony. (2011). *Real time vehicle speed prediction using a neural network traffic model.* International Joint Conference on Neural Networks (IJCNN), San Jose, CA, (pp. 2991-2996).

Parpinelli, Rafael S, & Lopes, Heitor S. (2011). New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation, 3*(1), 1-16.

Peng, BIAN. (2011). Combined prediction for traffic flow based on particle swarm optimization. *Journal of Chongqing Technology and Business University (Natural Science Edition), 1*, 15.

Pfoser, Dieter. (2008). Floating car data. *Encyclopedia of GIS*, 321-321.

Pham, DT, Ghanbarzadeh, A, Koc, E, Otri, S, Rahim, S, & Zaidi, M. (2006). *The bees algorithm-a novel tool for complex optimisation problems.* Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006), (pp. 454-459).

Piorkowski, Michal, Grossglauser, Matthias, & Papaioannou, Aristidis. (2006). *Mobile user navigation supported by WSAN: full-fledge demo of the SmartPark system.* 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc" 06).

Piorkowski, Michal, Raya, Maxim, Lugo, A Lezama, Papadimitratos, Panagiotis, Grossglauser, Matthias, & Hubaux, J-P. (2008). TraNS: realistic joint traffic and network simulator for VANETs. *ACM SIGMOBILE Mobile Computing and Communications Review, 12*(1), 31-33.

PTV, AG. (2006). VISUM homepage. Retrieved 4/7/2014, from http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/

Putha, Rahul, Quadrifoglio, Luca, & Zechman, Emily. (2012). Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions. *Computer‐Aided Civil and Infrastructure Engineering, 27*(1), 14-28.

Qualnet. (2011). Scalable network technologies. from www. qualnet. com

Qun, Chen. (2009). *Dynamic route guidance method based on particle swarm optimization algorithm.* Second International Conference on Intelligent Computation Technology and Automation (ICICTA'09) Changsha, Hunan, (pp. 267-270).

Rakha, H, & Van Aerde, M. (1995). Statistical analysis of day-to-day variations in real-time traffic flow data. *Transportation research record*, 1-27.

Raya, Maxim, Papadimitratos, Panos, & Hubaux, Jean-Pierre. (2006). Securing vehicular communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, 13*(LCA-ARTICLE-2006-015), 8-15.

Riley, George F. (2003). *The georgia tech network simulator.* Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research, (pp. 5-12).

Rumelhart, David E, Hinton, Geoffrey E, & Williams, Ronald J. (1988). Learning representations by back-propagating errors *Cognitive modeling*: MIT press.

Rybicki, Jedrzej, Pesch, Benjamin, Mauve, Martin, & Scheuermann, Björn. (2011). *Supporting cooperative traffic information systems through street-graph-based peer-to-peer networks.* 17th GI/ITG Conference on Communication in Distributed Systems (KiVS'11), Kiel, Germany, (Vol. 17, pp. 121-132).

Rybicki, Jedrzej, Scheuermann, Björn, Koegel, Markus, & Mauve, Martin. (2009). *PeerTIS: a peer-to-peer traffic information system.* Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking, (pp. 23-32).

Salehi-nezhad, Hojjat, & Farrahi-Moghaddam, Fereydoun. (2007). *An ant based algorithm approach to vehicle navigation.* First Joint Congress on Fuzzy and Intelligent Systems, (pp. 1-10).

Salehinejad, Hojjat, & Talebi, Siamak. (2008). *A new ant algorithm based vehicle navigation system: a wireless networking approach.* International Symposium on Telecommunications (IST), Tehran, (pp. 36-41).

Salehinejad, Hojjat, & Talebi, Siamak. (2010). Dynamic fuzzy logic-ant colony system-based route selection system. *Applied Computational Intelligence and Soft Computing, 2010*, 1-13. doi: 10.1155/2010/428270

Sanketh, Kumar P, Subbarao, Sanjay, & Jolapara, Kushal A. (2010). *I2V and V2V communication based VANET to optimize fuel consumption at traffic signals.* 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), Funchal, (pp. 1251-1255).

Schilling, Bjorn. (2005). Qualitative comparison of network simulation tools. *Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart*, 1-15.

Schmitt, Erick J, & Jula, Hossein. (2006). *Vehicle route guidance systems: Classification and comparison.* IEEE Intelligent Transportation Systems Conference (ITSC'06), Toronto, (pp. 242-247).

Schoonderwoerd, Ruud, Holland, Owen E, Bruten, Janet L, & Rothkrantz, Leon JM. (1997). Ant-based load balancing in telecommunications networks. *Adaptive behavior, 5*(2), 169-207.

Schrank, David, Eisele, Bill, & Lomax, Tim. (2012). TTI's 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System*.

Senge, Sebastian, & Wedde, Horst. (2010). *Bee inspired online vehicle routing in large traffic systems.* The Second International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2010), Lisbon, Portugal, (pp. 78-83).

Senge, Sebastian, & Wedde, Horst F. (2012a). *2-way evaluation of the distributed beejama vehicle routing approach.* IEEE Intelligent Vehicles Symposium (IV), Alcala de Henares, (pp. 205-210).

Senge, Sebastian, & Wedde, Horst F. (2012b). *Bee-inpired road traffic control as an example of swarm intelligence in cyber-physical systems.* 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Cesme, Izmir, (pp. 258-265).

Senge, Sebastian, & Wedde, Horst F. (2012c). Minimizing vehicular travel times using the multi-agent system beejama *Product-Focused Software Process Improvement* (pp. 335-349): Springer.

Sha, Wenjie, Kwak, Daehan, Nath, Badri, & Iftode, Liviu. (2013). *Social vehicle navigation: integrating shared driving experience into vehicle navigation*. Proceedings of the 14th Workshop on Mobile Computing Systems and Applications (pp. 1-6).

Shah, Sameena, Bhaya, Amit, Kothari, Ravi, & Chandra, Suresh. (2013). Ants find the shortest path: a mathematical proof. *Swarm Intelligence, 7*(1), 43-62.

Shen, Luou. (2008). *Freeway travel time estimation and prediction using dynamic neural networks* (PhD thesis, Florida International University).

Shi, Xi, Xu, Jianming, Xu, Yangsheng, & Song, Jingyan. (2005). *A simulation study on agent-network based route guidance system.* IEEE Intelligent Transportation Systems (pp. 248-253).

Shmygelska, Alena, & Hoos, Holger H. (2005). An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC bioinformatics, 6*(1), 30.

Sichitiu, Mihail L, & Kihl, Maria. (2008). Inter-vehicle communication systems: a survey. *IEEE Communications Surveys & Tutorials, 10*(2), 88-105.

Sikdar, Biplab. (2012). Comparison of broadcasting schemes for infrastructure to vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, *13*(2), 492-502.

Sjöberg Bilstrup, Katrin. (2009). *Predictable and scalable medium access control for vehicular ad hoc networks* (PhD thesis, Halmstad University).

Socha, Krzysztof, & Blum, Christian. (2007). An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. *Neural Computing and Applications, 16*(3), 235-247.

Socha, Krzysztof, Sampels, Michael, & Manfrin, Max. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art *Applications of evolutionary computing* (pp. 334-345): Springer.

Solnon, Christine, & Fenet, Serge. (2006). A study of ACO capabilities for solving the maximum clique problem. *Journal of Heuristics, 12*(3), 155-180.

Sommer, Christoph, Eckhoff, David, German, Reinhard, & Dressler, Falko. (2011). *A computationally inexpensive empirical model of IEEE 802.11 p radio shadowing in urban environments.* Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS), Bardonecchia, (pp. 84-90).

Sommer, Christoph, Krul, Robert, German, Reinhard, & Dressler, Falko. (2010). *Emissions vs. travel time: simulative evaluation of the environmental impact of ITS.* 71st IEEE Vehicular Technology Conference, (VTC-Spring), Taipei, (pp. 1-5).

Stabler, Ben. (2006). Shapefiles: read and write ESRI shapefiles. *R package version 0.6*.

Stützle, Thomas, & Hoos, Holger H. (2000). MAX–MIN ant system. *Future generation computer systems, 16*(8), 889-914.

Sun, Min-Te, Feng, Wu-Chi, Lai, Ten-Hwang, Yamada, Kentaro, Okada, Hiromi, & Fujimura, Kikuo. (2000). *GPS-based message broadcast for adaptive inter-vehicle communications.* 52nd IEEE Vehicular Technology Conference (IEEE-VTS Fall VTC) Boston, MA, (Vol. 6, pp. 2685-2692).

Sur, Chiranjib, Sharma, Sanjeev, & Shukla, Anupam. (2012). *Analysis & modeling multi-breeded Mean-Minded ant colony optimization of agent based Road Vehicle Routing Management.* International Conference for Internet Technology And Secured Transactions, London, (pp. 634-641).

Suson, A. (2010). *Dynamic routing using ant-based control* (Master thesis, Delft University of Technology, Netherland).

Taleb, Tarik, Sakhaee, Ehssan, Jamalipour, Abbas, Hashimoto, Kazuo, Kato, Nei, & Nemoto, Yoshiaki. (2007). A stable routing protocol to support ITS services in VANET networks. *IEEE Transactions on Vehicular Technology, 56*(6), 3337-3347.

Tatomir, B, & Rothkrantz, LJM. (2005). H-ABC: a scalable dynamic routing algorithm. *Recent Advances in Artificial Life, 5*, 8.

Tatomir, Bogdan, Kroon, Ronald, & Rothkrantz, Leon. (2004). Dynamic routing in traffic networks using AntNet. *Ant Colony Optimization and Swarm Intelligence* (pp. 424-425): Springer.

Tatomir, Bogdan, & Rothkrantz, Leon. (2004). *Dynamic traffic routing using ant based control.* IEEE International Conference on Systems, Man and Cybernetics, (Vol. 4, pp. 3970-3975).

Tatomir, Bogdan, & Rothkrantz, Leon. (2006). *Hierarchical routing in traffic using swarm-intelligence.* IEEE Intelligent Transportation Systems Conference (ITSC'06), Toronto, (pp. 230-235).

Tatomir, Bogdan, Rothkrantz, Leon JM, & Suson, Adriana-Camelia. (2009). *Travel time prediction for dynamic routing using ant based control.* Proceedings of the 2009 Winter Simulation Conference (WSC), Austin, TX, (pp. 1069-1078).

Teodorović, Dušan. (2008). Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C: Emerging Technologies, 16*(6), 651-667.

Teodorović, Dušan, & Dell'Orco, Mauro. (2005). *Bee colony optimization–a cooperative learning approach to complex transportation problems.* Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini–EURO Conference and 10th Meeting of EWGT, (pp. 51-60).

Teodorović, Dušan, & Dell'Orco, Mauro. (2008). Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization. *Transportation Planning and Technology, 31*(2), 135-152.

Teodorovic, Dusan, Edara, Praveen, & Via, Charles E. (2005). *Highway space inventory control system.* 16th International Symposium on Transportation and Traffic Theory: Flow, Dynamics and Human Interaction, College Park Maryland, United States, (pp. 43-62).

Terr, David. (2004). Weighted mean. *From MathWorld-A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/WeightedMean.html, 58,* 13918-13927.

Theraulaz, Guy, & Bonabeau, Eric. (1999). A brief history of stigmergy. *Artificial life, 5*(2), 97-116.

Tong, CO, & Wong, SC. (2000). A predictive dynamic traffic assignment model in congested capacity-constrained road networks. *Transportation Research Part B: Methodological, 34*(8), 625-644.

Tonguz, Ozan K. (2011). Biologically inspired solutions to fundamental transportation problems. *IEEE Communications Magazine, 49*(11), 106-115.

Toth, Paolo, & Vigo, Daniele. (2001). *The vehicle routing problem: Society for Industrial and Applied Mathematics*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, USA, (pp. 1-6).

Tubaishat, Malik, Shang, Yi, & Shi, Hongchi. (2007). *Adaptive traffic light control with wireless sensor networks.* Proceedings of IEEE Consumer Communications and Networking Conference, Las Vegas, Nevada, (pp. 187-191).

Ulungu, EL, & Teghem, J. (1991). *Multi-objective shortest path problem: A survey*. Proceedings of the International Workshop on Multicriteria Decision Making: Methods–Algorithms–Applications, Liblice, Czechoslovakia, (pp. 176-188).

Van Lint, JWC, Hoogendoorn, SP, & van Zuylen, Henk J. (2005). Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, *13*(5), 347-369.

Varga, Andras. (2010). OMNeT++ *Modeling and Tools for Network Simulation* (pp. 35-59): Springer.

Walsh, K, & Sirer, EG. (2003). A staged network simulator (SNS). *Computer Science Department, Cornell University*.

Wang, Shie-Yuan, Chou, CL, Huang, CH, Hwang, CC, Yang, ZM, Chiou, CC, & Lin, CC. (2003). The design and implementation of the NCTUns 1.0 network simulator. *Computer networks*, *42*(2), 175-197.

Wardrop, John Glen. (1952). Road paper. some theoretical aspects of road traffic research. *ICE Proceedings: Engineering Divisions*, *1*(3), 325-362.

WAZE. (2015). Waze official website. Retrieved 18/5/2015, from https://www.waze.com/

Wedde, Horst F, & Farooq, Muddassar. (2005). *The wisdom of the hive applied to mobile ad-hoc networks.* Proceedings 2005 IEEE Swarm Intelligence Symposium (SIS), (pp. 341-348).

Wedde, Horst F, Farooq, Muddassar, & Zhang, Yue. (2004). BeeHive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior. *Ant colony optimization and swarm intelligence* (pp. 83-94): Springer.

Wedde, Horst F, Lehnhoff, Sebastian, van Bonn, Bernhard, Bay, Zeynep, Becker, Sven, Bottcher, S, . . . Lazarescu, Anca M. (2007). *A novel class of multi-agent algorithms for highly dynamic transport planning inspired by honey bee behavior*. IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Patras, (pp. 1157-1164).

Wedde, Horst F, & Senge, Sebastian. (2013). BeeJamA: a distributed, self-adaptive vehicle routing guidance approach. *IEEE Transactions on Intelligent Transportation Systems*, *14*(4), 1882-1895.

Wedel, Jan W, Schunemann, B, & Radusch, Ilja. (2009). *V2X-based traffic congestion recognition and avoidance.* 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN), Kaohsiung, (pp. 637-641).

Wegener, Axel, Piórkowski, Michał, Raya, Maxim, Hellbrück, Horst, Fischer, Stefan, & Hubaux, Jean-Pierre. (2008). *TraCI: an interface for coupling road traffic and network simulators.* Proceedings of the 11th communications and networking simulation symposium, (pp. 155-163).

Wen, Feng, & Gen, Mitsuo. (2008). *A genetic-based clustering approach to traffic network design for car navigation system.* IEEE International Conference on Systems, Man and Cybernetics (SMC), Singapore, (pp. 1688-1693).

Wen, Feng, Gen, Mitsuo, & Yu, Xinjie. (2011). A new multiobjective genetic algorithm for route selection. *IEEJ Transactions on Electronics, Information and Systems, 131*(3), 619-625.

Wen, Feng, & Lin, Chiming. (2010). Multiobjective route selection model and its soving method based on genetic algorithm. *international journal of information system for logistics and management, 5*(2), 1-8.

Wetteroth, Debbra. (2001). *OSI reference model for telecommunications*: McGraw-Hill Professional.

Weyns, Danny, Holvoet, Tom, & Helleboogh, Alexander. (2007). *Anticipatory vehicle routing using delegate multi-agent systems.* IEEE Intelligent Transportation Systems Conference (ITSC), Seattle, WA, (pp. 87-93).

Wilensky, Uri. (1999). NetLogo: center for connected learning and computer-based modeling. *Northwestern University*.

Wischoff, L, Ebner, André, Rohling, Hermann, Lott, Matthias, & Halfmann, Rüdiger. (2003). *SOTIS-a self-organizing traffic information system.* 57th IEEE Semiannual Vehicular Technology Conference (VTC-Spring), (Vol. 4, pp. 2442-2446).

Wu, Lei, Yang, Licai, Liu, Haiqing, & Zhang, Yao. (2014). Bee inspired zonal vehicle routing algorithm in urban traffic. *TELKOMNIKA Indonesian Journal of Electrical Engineering, 12*(9), 6699-6710.

Wu, Yan-Jing, & Sung, Wei-Cheng. (2010). *A dynamic navigation scheme for vehicular ad hoc networks.* Sixth International Conference on Networked Computing and Advanced Information Management (NCM), Seoul, (pp. 231-235).

Xu, Qing-zheng, & Ke, Xi-zheng. (2008). Genetic algorithm analysis for shortest path. *Computer Engineering and Design, 6*, 1507-1509.

Xu, Zhihong, Hou, Xiangdan, & Sun, Jizhou. (2003). *Ant algorithm-based task scheduling in grid computing.* IEEE Canadian Conference on Electrical and Computer Engineering (IEEE CCECE), (Vol. 2, pp. 1107-1110).

Yamashita, Tomohisa, Izumi, Kiyoshi, & Kurumatani, Koichi. (2004). *Car navigation with route information sharing for improvement of traffic efficiency.* 7th International IEEE Conference on Intelligent Transportation Systems, (pp. 465-470).

Yang, Lin, Guo, Jinhua, & Wu, Ying. (2008). *Channel adaptive one hop broadcasting for VANETs.* 11th International IEEE Conference on Intelligent Transportation Systems (ITSC) Beijing, (pp. 369-374).

Yang, Xin-She. (2005). Engineering optimizations via nature-inspired virtual bee algorithms. *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach* (pp. 317-323): Springer.

Yang, Xin-She. (2009). Firefly algorithms for multimodal optimization *Stochastic algorithms: foundations and applications* (pp. 169-178): Springer.

Yang, Xin-She. (2010a). *Nature-inspired metaheuristic algorithms*: Luniver press.

Yang, Xin-She. (2010b). A new metaheuristic bat-inspired algorithm *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74): Springer.

Yang, Xin-She. (2011a). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation, 3*(5), 267-274.

Yang, Xin-She. (2011b). Chaos-enhanced firefly algorithm with automatic parameter tuning. *International Journal of Swarm Intelligence Research (IJSIR), 2*(4), 1-11.

Yang, Xin-She, Cui, Zhihua, Xiao, Renbin, Gandomi, Amir Hossein, & Karamanoglu, Mehmet. (2013). *Swarm intelligence and bio-inspired computation: theory and applications*: Newnes, (pp. 1-450).

Yang, Xin-She, & Deb, Suash. (2009). *Cuckoo search via Lévy flights.* World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, (pp. 210-214).

Yang, Xin-She, Deb, Suash, & Fong, Simon. (2011). Accelerated particle swarm optimization and support vector machine for business optimization and applications. *Networked digital technologies* (pp. 53-66): Springer.

Yang, Xin-She, & Gandomi, Amir Hossein. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations, 29*(5), 464-483.

Yang, Zhaosheng, Zhang, Lin, Wang, Jian, Wang, Yuan, Guan, Qing, & Feng, Jinqiao. (2006). *Design of intelligent in-vehicle navigation systems for dynamic route guidance with real-time information.* IEEE International Conference on Vehicular Electronics and Safety (ICVES), Beijing, (pp. 184-188).

Yousefi, Pirayeh, & Zamani, Roghayeh. (2013). The optimal routing of cars in the car navigation system by taking the combination of divide and conquer method and ant colony algorithm into consideration. *International Journal of Machine Learning and Computing, 3,* 44-48.

Yu, Haicong, & Lu, Feng. (2012). A multi-modal route planning approach with an improved genetic algorithm. *Advances in Geo-Spatial Information Science* (pp. 193): CRC Press.

Yue, Hao, & Shao, Chunfu. (2007). *Study on the application of A* shortest path search algorithm in dynamic urban traffic.* Third International Conference on Natural Computation (ICNC), Haikou, (pp. 463-469).

Zhan, F Benjamin. (1997). Three fastest shortest path algorithms on real road networks: Data structures and procedures. *Journal of geographic information and decision analysis*, *1*(1), 69-82.

Zhan, F Benjamin, & Noon, Charles E. (1998). Shortest path algorithms: an evaluation using real road networks. *Transportation Science, 32*(1), 65-73.

Zhang, Jiang, Ziliaskopoulos, Athanasios K, Wen, Ning, & Berry, Randall A. (2005). *Design and implementation of a vehicle-to-vehicle based traffic information system.* IEEE Intelligent Transportation Systems, (pp. 473-477).

Zhang, Wei, Zhang, Zi, & Xu, Jianmin. (2006). *Dynamic route guidance using neurodynamic programming.* Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), Jinan, (pp. 1177-1181).

Zhang, Xiaoyan, & Rice, John A. (2003). Short-term travel time prediction. *Transportation Research Part C: Emerging Technologies, 11*(3), 187-210.

Zhang, Xuedan, Hong, Jun, Fan, Shuai, Wei, Zhongya, Cao, Jiannong, & Ren, Yong. (2007). *A novel real-time traffic information system based on wireless mesh networks.* IEEE Intelligent Transportation Systems Conference (ITSC), Seattle, WA, (pp. 618-623).

Zhou, Binbin, Cao, Jiannong, Zeng, Xiaoqin, & Wu, Hejun. (2010). *Adaptive traffic light control in wireless sensor network-based intelligent transportation system.* 72nd IEEE Vehicular Technology Conference Fall (VTC-Fall), Ottawa, ON, (pp. 1-5).

Zhou, Lidong, & Haas, Zygmunt J. (1999). Securing ad hoc networks. *IEEE Network, 13*(6), 24-30.

Zhu, Kun, Niyato, Dusit, Wang, Ping, Hossain, Ekram, & In Kim, Dong. (2011). Mobility and handoff management in vehicular networks: a survey. *Wireless communications and mobile computing, 11*(4), 459-476.

Zong, Xinlu, Xiong, Shengwu, Fang, Zhixiang, & Li, Qiuping. (2010). *Multi-ant colony system for evacuation routing problem with mixed traffic flow.* IEEE Congress on Evolutionary Computation (CEC), Barcelona, (pp. 1-6).

List of Publications and Papers Presented

- Mohammad Reza Jabbarpour, Hossein Malakooti, Ali Jalooli and Rafidah Md Noor, "A Dynamic Vehicular Traffic Control Using Ant Colony and Traffic Light Optimization", *International Conference on Systems Science (ICSS 2013),* September 10-12, 2013, Wroclaw, Poland.
- Mohammad Reza Jabbarpour, Hossein Malakooti, Rafidah Md Noor, Nor Badrul Anuar and Norazlina Khamis, "Ant Colony Optimisation for Vehicle Traffic Systems: Applications and Challenges", *International Journal of Bio-Inspired Computation (IF = 1.682, Q1)*, 2014, Vol. 6, No. 1, pp. 32-56.
- Mohammad Reza Jabbarpour, Ali Jalooli, Erfan Shaghaghi, Rafidah Md Noor, Leon Rothkrantz, Rashid Hafeez Khokhar and Nor Badrul Anuar, "Ant-based Vehicle Congestion Avoidance System using Vehicular Networks", *Engineering Applications of Artificial Intelligence (IF = 1.962, Q1),* 2014, Vol. 36, pp. 303-319.
- Mohammad Reza Jabbarpour and Rafidah Md Noor, "A Novel Green Car Navigation System for Vehicle Congestion Avoidance using Ant-based Algorithm", *Copyright*, 2014, No: LY2014001218.
- Mohammad Reza Jabbarpour, Rafidah Md Noor and Rashid Hafeez Khokhar, "Green Vehicle Traffic Routing System using Ant Colonial-based Algorithm", *Journal of Network and Computer Applications (IF = 1.772, Q1)*, 2014, (Under review, Submitted Nov. 2014)

Appendixes

Appendix A. A part of *map.nod.xml* file


```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
 generated on 07/16/14 22:49:12 by SUMO netconvert Version 0.21.0
<?xml version="1.0" encoding="UTF-8"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/netconvertConfiguration.xsd">

   <input>
      <osm-files value="map.osm"/>
   </input>

   <output>
      <plain-output-prefix value="map"/>
   </output>

   <projection>
      <proj.utm value="true"/>
   </projection>

</configuration>
  -->
- <nodes version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/nodes_file.xsd">
      <location netOffset="-799395.39,-344143.98" convBoundary="0.00,0.00,4676.13,5795.46" origBoundary="101.693647,3.110066,101.735708,3.174000" projParameter="+proj=utm +zone=47 +ellps=WGS84 +datum=WGS84 +units=m +no_defs" />
          <node id="1154363098" x="758.85" y="3362.63" type="priority" />
          <node id="1198097101" x="1875.59" y="5385.81" type="unregulated" />
          <node id="1198097121" x="1226.63" y="5151.44" type="priority" />
          <node id="1198097187" x="874.22" y="4991.82" type="priority" />
          <node id="1198097212" x="884.51" y="5008.55" type="priority" />
          <node id="1198097217" x="876.10" y="4959.04" type="priority" />
          <node id="1198097243" x="1209.71" y="5150.53" type="priority" />
          <node id="1198097245" x="1018.37" y="5218.74" type="priority" />
          <node id="1198097281" x="565.85" y="4668.94" type="unregulated" />
          <node id="1198097317" x="1275.06" y="5164.82" type="priority" />
         .

         .

         .
          <node id="92282714" x="2072.53" y="4500.06" type="priority" />
          <node id="92291222" x="2113.75" y="4973.59" type="priority" />
          <node id="92291233" x="2597.38" y="5011.00" type="priority" />
          <node id="92291234" x="2573.10" y="5053.57" type="priority" />
          <node id="92291239" x="2586.48" y="5164.59" type="right_before_left" />
          <node id="92291247" x="2309.61" y="5163.87" type="priority" />
          <node id="92291250" x="2219.57" y="5089.57" type="priority" />
          <node id="92291251" x="2193.37" y="5071.58" type="priority" />
          <node id="92291253" x="2128.50" y="5053.39" type="right_before_left" />
</nodes>
```

Appendix B. A part of *map.edg.xml* file


```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
 generated on 07/16/14 22:49:12 by SUMO netconvert Version 0.21.0
<?xml version="1.0" encoding="UTF-8"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/netconvertConfiguration.xsd">

    <input>
        <osm-files value="map.osm"/>
    </input>

    <output>
        <plain-output-prefix value="map"/>
    </output>

    <projection>
        <proj.utm value="true"/>
    </projection>

</configuration>
  -->
- <edges version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/edges_file.xsd">
  <edge id="-103774410" from="1198097245" to="1198097334" priority="4" type="highway.residential" numLanes="1" speed="13.89" shape="1018.37,5218.74 1041.28,5163.44 1043.14,5131.66" />
  <edge id="-103777529" from="1756919004" to="1198144249" priority="4" type="highway.residential" numLanes="1" speed="13.89" />
  <edge id="-10589714#0" from="1814470613" to="34419737" priority="6" type="highway.tertiary" numLanes="1" speed="22.22" shape="2153.65,4606.48 2185.74,4675.14 2203.21,4707.81 2217.73,4731.27" />
  <edge id="-10589714#1" from="1814470575" to="1814470613" priority="6" type="highway.tertiary" numLanes="1" speed="22.22" />
  <edge id="-10589714#2" from="92278903" to="1814470575" priority="6" type="highway.tertiary" numLanes="1" speed="22.22" />
  <edge id="-10589714#3" from="28918012" to="92278903" priority="6" type="highway.tertiary" numLanes="1" speed="22.22" shape="2101.18,4455.66 2102.56,4485.01 2105.04,4500.82" />
  .
  .
  .
  <edge id="53154470" from="672688243" to="1722609833" priority="6" type="highway.secondary_link" numLanes="1" speed="22.22" shape="2279.62,3487.33 2302.68,3475.23 2321.75,3468.95 2349.94,3470.57 2374.56,3480.58" spreadType="center" />
  <edge id="99873605" from="478424497" to="478501482" priority="4" type="highway.residential" numLanes="1" speed="13.89" spreadType="center" />
  <edge id="99873607#0" from="478549108" to="1821025834" priority="4" type="highway.residential" numLanes="1" speed="13.89" shape="780.17,3808.90 791.05,3745.70 794.93,3674.75" />
  <edge id="99873607#1" from="1821025834" to="475482687" priority="4" type="highway.residential" numLanes="1" speed="13.89" />
  <edge id="99873611#0" from="478450158" to="478911699" priority="4" type="highway.residential" numLanes="1" speed="13.89" shape="1705.53,3029.63 1683.35,3055.18 1585.32,3129.21 1491.54,3202.05" />
  <edge id="99873611#1" from="478911699" to="478424497" priority="4" type="highway.residential" numLanes="1" speed="13.89" />
</edges>
```

Appendix C. A part of *map.con.xml* file


```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
 generated on 07/16/14 22:49:12 by SUMO netconvert Version 0.21.0
<?xml version="1.0" encoding="UTF-8"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/netconvertConfiguration.xsd">

    <input>
        <osm-files value="map.osm"/>
    </input>

    <output>
        <plain-output-prefix value="map"/>
    </output>

    <projection>
        <proj.utm value="true"/>
    </projection>

</configuration>
 -->
<connections version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/connections_file.xsd">
  <connection from="-103774410" to="39901847#4" fromLane="0" toLane="1" />
  <connection from="-103774410" to="103774410" fromLane="0" toLane="0" />
  <connection from="-103777529" to="-23766090#1" fromLane="0" toLane="0" />
  <connection from="-103777529" to="23766090#2" fromLane="0" toLane="0" />
  <connection from="-103777529" to="103777529" fromLane="0" toLane="0" />
  <connection from="-10589714#0" to="5116003#0" fromLane="0" toLane="0" />
  <connection from="-10589714#0" to="10589714#0" fromLane="0" toLane="0" />
  <connection from="-10589714#1" to="-170293888" fromLane="0" toLane="0" />
  <connection from="-10589714#1" to="-10589714#0" fromLane="0" toLane="0" />
  <connection from="-10589714#1" to="10589714#1" fromLane="0" toLane="0" />
  <connection from="-10589714#2" to="-170293889" fromLane="0" toLane="0" />
  <connection from="-10589714#2" to="-10589714#1" fromLane="0" toLane="0" />
  <connection from="-10589714#2" to="10589714#2" fromLane="0" toLane="0" />
  <connection from="-10589714#3" to="-10589714#2" fromLane="0" toLane="0" />
  <connection from="-10589714#3" to="198263663" fromLane="0" toLane="0" />
.

.

.
  <connection from="99873607#0" to="99873607#1" fromLane="0" toLane="0" />
  <connection from="99873607#0" to="-170998159" fromLane="0" toLane="0" />
  <connection from="99873607#0" to="-99873607#0" fromLane="0" toLane="0" />
  <connection from="99873607#1" to="258714765#1" fromLane="0" toLane="0" />
  <connection from="99873607#1" to="-99873607#1" fromLane="0" toLane="0" />
  <connection from="99873611#0" to="99873611#1" fromLane="0" toLane="0" />
  <connection from="99873611#0" to="39922262#0" fromLane="0" toLane="0" />
  <connection from="99873611#0" to="-99873611#0" fromLane="0" toLane="0" />
  <connection from="99873611#1" to="99873605" fromLane="0" toLane="0" />
  <connection from="99873611#1" to="-99873611#1" fromLane="0" toLane="0" />
</connections>
```

Appendix D. *ant_pkt.h* file

```
#ifndef __ant_pkt_h__
#define __ant_pkt_h__

#include <packet.h>
#include <list>
#include "AVCAF_common.h"

#define FANT 0x01
#define BANT 0x02
#define ANT_SIZE 9
#define HDR_ANT_PKT(p) hdr_ant_pkt::access(p)

struct memory{
        nsaddr_t node_addr;
        double trip_time;
};

struct hdr_ant_pkt {
    u_int8_t pkt_type_;
        nsaddr_t pkt_src_;
        nsaddr_t pkt_src_seg_;
        nsaddr_t pkt_dst_;
        nsaddr_t pkt_dst_seg_;
        u_int16_t pkt_len_;
        u_int8_t pkt_seq_num_;
        double pkt_start_time_;
        struct memory pkt_memory_[MAX_NUM_NODES];
        int pkt_mem_size_;

        inline nsaddr_t& pkt_src() {return pkt_src_;}
        inline nsaddr_t& pkt_src_seg() {return pkt_dst_seg_;}
        inline nsaddr_t& pkt_dst() {return pkt_src_;}
        inline nsaddr_t& pkt_dst_seg() {return pkt_dst_seg_;}
        inline u_int16_t& pkt_len() {return pkt_len_;}
        inline u_int8_t& pkt_seq_num() {return pkt_seq_num_;}
        inline double& pkt_start_time() {return pkt_start_time_;}
        inline int& pkt_mem_size() {return pkt_mem_size_;}
        inline u_int8_t& pkt_type() {return pkt_type_;}

        static int offset_;
        inline static int& offset() {return offset_;}

        inline static hdr_ant_pkt* access(const Packet *p) {
                return (hdr_ant_pkt*)p->access(offset_);
        }
};

#endif
```

Appendix E. *AVCAF_Intra_ST.h* file

```
#ifndef __AVCAF_Intra_ST_h__
#define __AVCAF_Intra_ST_h__

#include <trace.h>
#include <map>
#include <string>
#include <vector>
#include <classifier-port.h>
#include <random.h>

#include "ant_pkt.h"
#include "AVCAF_common.h"

struct pheromone {
	nsaddr_t neighbor;
	double phvalue;
};

typedef std::vector<struct pheromone> pheromone_matrix;
typedef std::map<nsaddr_t, pheromone_matrix> Intra_ST;
typedef std::vector<nsaddr_t> sameph_t;

class AVCAF_Intra_ST {
	Intra_ST rt_;

	public:

		AVCAF_Intra_ST() {

		void add_entry(nsaddr_t destination, nsaddr_t neighbor, double phvalue);
		void print();
		nsaddr_t calc_destination(nsaddr_t source, int source_seg);
		nsaddr_t calc_next(nsaddr_t source, nsaddr_t destination, nsaddr_t parent, int
source_seg, int destination_seg);
		void update(nsaddr_t destination, nsaddr_t neighbor, int source_seg, int
destination_seg);
};

#endif
```

Appendix F. *AVCAF_ant.h* file

```
#ifndef __ AVCAF_ant _h__
#define __ AVCAF_ant _h__

#include <agent.h>
#include <node.h>
#include <packet.h>
#include <ip.h>
#include <trace.h>
#include <timer-handler.h>
#include <random.h>
#include <classifier-port.h>
#include <tools/rng.h>

#include "trace/cmu-trace.h"
#include "tools/queue-monitor.h"
#include "queue/drop-tail.h"

#include "ant_pkt.h"
```

```
#include "AVCAF_common.h"
#include " AVCAF_Intra_ST.h"

#include <map>
#include <vector>
#include <list>

class AVCAF_ant;

class Ant_timer: public TimerHandler {
	public:
		Ant_timer(AVCAF_ant* agent) : TimerHandler() {
			agent_ = agent;
		}
	protected:
		AVCAF_ant * agent_;
		virtual void expire(Event* e);
};

class AVCAF_ant: public Agent {

	friend class Ant_timer;

	nsaddr_t  ra_addr_;
	AVCAF_Intra_ST  Intra_ST_;
	u_int8_t ant_seq_num_;

	double xi_;
	double lambda_;
	double alpha_;
	double beta_;
	double rho_;
	double gamma_;
	int num_paths_;

	protected:
		PortClassifier* dmux_;
		Trace* logtarget_;
		Ant_timer ant_timer_;

		void reset_ant_timer();
		void send_ant_pkt();
		void recv_ant_pkt(Packet*);
		void create_backward_ant_pkt(Packet*);
		void forward_ant_pkt(Packet*);
		void backward_ant_pkt(Packet*);
		void memorize(Packet*);
		void update_Intra_ST(Packet*);
		void print_neighbors();
		void add_Neighbor(Node* node1, Node* node2);

		void initialize_Intra_ST();

	public:
		ANCAS_ant(nsaddr_t);
		int command(int , const char*const*);
		void recv(Packet*, Handler*);
};

#endif
```

Appendix G. *AVCAF_common.h* file

```
#ifndef __ AVCAF_common_h__
#define __ AVCAF_common_h__

#include <agent.h>
#include <node.h>
#include <packet.h>
#include <ip.h>
#include <trace.h>
#include <timer-handler.h>
#include <random.h>
#include <classifier-port.h>
#include <tools/rng.h>

#include "trace/cmu-trace.h"
#include "tools/queue-monitor.h"
#include "queue/drop-tail.h"

#define CURRENT_TIME Scheduler::instance().clock()

#define file_rtable "Intra_ST.txt"

#define DEBUG 0

#define WIN_LEN 300

int get_num_neighbors(nsaddr_t node_addr);

#endif
```

Appendix H. *AVCAF_common.cc* file

```
#include "AVCAF_common.h"

Int get_num_neighbors(nsaddr_t node_addr) {
	int count = 0;
	Node *nd = nd->get_node_by_address(node_addr);
	neighbor_list_node* nb = nd->neighbor_list_;
	while(nb != NULL) {
		count ++;
		nb = nb->next;
	}
	return count;
}
```