An Ant Inspired Dynamic Traffic Assignment for VANETs:

Early Notification of Traffic Congestion and Traffic Incidents

by

Wilmer Arellano

A Dissertation Submitted to the Faculty of

The College of Engineering and Computer Science

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

May 2016

An Ant Inspired Dynamic Traffic Assignment for VANETs:

Early Notification of Traffic Congestion and Traffic Incidents

by

Wilmer Arellano

This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Imad Mahgoub, Department of Computer and Electrical Engineering and Computer Science, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

SUPERVISORY COMMITTEE:

Imad Mahgoub, Ph.D.
Dissertation Advisor

Mohammad Ilyas, Ph.D.

Mihaela Cardei, Ph.D.

Ed Callaway, Ph.D.

Nurgun Erdol, Ph.D.
Chair, Department of Computer and Electrical Engineering and Computer Science

Mohammad Ilyas, Ph.D.
Dean, College of Engineering and Computer Science

Deborah L. Floyd, Ed.D.
Dean, Graduate College

05/20/2016
Date

ACKNOWLEDGEMENTS

It is with the most sincere feeling that I would like to express my gratitude to the members of my supervisory committee for all their assertive guidance and understanding. Special thanks to my advisor Dr. Imad Mahgoub whom with patience always shared his technical and human wisdom. Last but not least I would like express my deep gratitude to Florida Atlantic University for the opportunity given to me to obtain a PhD degree in this magnificent institution, to the faculty and employees who directly impacted this journey, as well as to those who indirectly or invisibly made it possible.

ABSTRACT

Author:	Wilmer Arellano
Title:	An Ant Inspired Dynamic Traffic Assignment for VANETs: Early Notification of Traffic Congestion and Traffic Incidents
Institution:	Florida Atlantic University
Dissertation Advisor:	Dr. Imad Mahgoub
Degree:	Doctor of Philosophy
Year:	2016

Vehicular Ad hoc NETworks (VANETs) are a subclass of Mobile Ad hoc NETworks and represent a relatively new and very active field of research. VANETs will enable in the near future applications that will dramatically improve roadway safety and traffic efficiency. There is a need to increase traffic efficiency as the gap between the traveled and the physical lane miles keeps increasing. The Dynamic Traffic Assignment problem tries to dynamically distribute vehicles efficiently on the road network and in accordance with their origins and destinations. We present a novel dynamic decentralized and infrastructure-less algorithm to alleviate traffic congestions on road networks and to fill the void left by current algorithms which are either static, centralized, or require infrastructure. The algorithm follows an online approach that seeks stochastic user equilibrium and assigns traffic as it evolves in real time, without prior knowledge of the

traffic demand or the schedule of the cars that will enter the road network in the future. The Reverse Online Algorithm for the Dynamic Traffic Assignment inspired by Ant Colony Optimization for VANETs follows a metaheuristic approach that uses reports from other vehicles to update the vehicle's perceived view of the road network and change route if necessary. To alleviate the broadcast storm spontaneous clusters are created around traffic incidents and a threshold system based on the level of congestion is used to limit the number of incidents to be reported. Simulation results for the algorithm show a great improvement on travel time over routing based on shortest distance. As the VANET transceivers have a limited range, that would limit messages to reach at most 1,000 meters, we present a modified version of this algorithm that uses a rebroadcasting scheme. This rebroadcasting scheme has been successfully tested on roadways with segments of up to 4,000 meters. This is accomplished for the case of traffic flowing in a single direction on the roads. It is anticipated that future simulations will show further improvement when traffic in the other direction is introduced and vehicles travelling in that direction are allowed to use a store carry and forward mechanism.

DEDICATION

I would like to dedicate this paper to my family, for their support, and understanding about the time I did not dedicate to them.

An Ant Inspired Dynamic Traffic Assignment for VANETs:

Early Notification of Traffic Congestion and Traffic Incidents

List of Tables

List of Figures

1. INTRODUCTION

Vehicular Ad hoc NETworks (VANETs) are a subclass of Mobile Ad hoc NETwork (MANETs) and represent a relatively new and very active field of research. Some particular characteristics of VANETs that make them unique are high-speed mobility, driver behavior that is dependent on personality traits, and mobility constraints as cars move on roadways with set boundaries. VANETs will enable in the near future applications that will dramatically improve traffic flow in the highways and improve significantly the associated ecological impact. According to the Federal Highway Administration (FHWA) [1], the U.S. highway network was near completion by the late 1980s; there has been little construction of new roads and highways since the number of lane miles has been increased mainly by adding additional lanes to carry more vehicles. From 1985 to 2006 the lane miles increased from 8 to 8.4 million while the Vehicle Miles Traveled doubled during the same period. From these figures and the familiar transit congestions that we face almost daily, we can infer that a new and intelligent approach is needed when optimizing road usage. In the United States, the federal Intelligent Transportation Systems (ITS), by means of the ITS Joint Program Office (JPO), leads research activities that focus on intelligent vehicles, intelligent infrastructure and the creation of an intelligent transportation system through integration with and between these two components. The ITS JPO is an element of the Research and Innovative Technology Administration (RITA), a unit of the U.S. Department of Transportation (DOT). RITA coordinates the U.S. DOT research programs. It is in charge of advancing the deployment of cross-cutting technologies to improve the Nation's transportation system. It is mentioned in ITS's Strategic Research Plan, 2010–2014 (Progress Update, 2012) [2] that vehicle fuel utilization and the resulting tailpipe emissions are the single largest human-made source of carbon dioxide, nitrous oxide, and methane. The Texas A&M Transportation Institute (TTI) is a higher-education affiliated transportation research agency. According to the TTI's 2012 URBAN MOBILITY REPORT [3], in 2011 losses related to traffic congestion are more than $121 billion annually, resulting from 5.5 billion lost hours and 2.9 billion gallons of wasted fuel.

Traffic assignment tries to distribute vehicles efficiently on the road network and in accordance with their origins and destinations [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Ant Colony Optimization (ACO) is a metaheuristic useful for obtaining minimum cost paths [19], [20], [21], [22], [23] [24], [25], [26]. This metaheuristic has been applied successfully in different branches of engineering [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37]. ACO has gained popularity recently for traffic assignment and many algorithms have been inspired by this metaheuristic [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49].

The events that will take place in the automotive industry in relation to VANETs and its ability to enable the automation of many processes in vehicles will have a technological impact of magnitude similar to the deployment of the High Definition Television System. However, it will have an unprecedented social impact because of the huge monetary savings and the positive impact on the environment due to fuel economy as traffic congestions decrease. This would probably be the first area of massive smart interaction between humans and machines. It is expected that, by providing solutions in the area of

An Ant Inspired Dynamic Traffic Assignment for VANETs: and Early Notification of Traffic Congestion and Traffic Incidents, we are helping to pave this road.

1.1 Traffic Assignment

There are two different models that can be identified when investigating road network transit problems: the *transportation planning models* and the *traffic flow models* [13]. The first one deals with modeling decisions made by individuals who use the roads and the approaches used by researchers to optimize traffic, while the second one deals with modeling the physical propagation of traffic flows. When selecting the best routes in *traffic assignment* it is typical to assign a cost based on distance and/or time to the different edges and then use a shortest path algorithm such as Dijkstra's algorithm [50]. Wardrop [4], established two equilibrium criteria relating to the traffic flow from a given *origin* to a certain *destination* by means of a number of available routes:

1. *User equilibrium assignment* (UE). The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.

2. *System Optimum Assignment* (SO). The average journey time is a minimum.

These criteria are also referred to in the literature as *deterministic user equilibrium assignment* (D-UE) and *deterministic system Optimum Assignment* (D-SO). We could say that in D-UE all decisions are made in an egoistic and rational way, and all users have knowledge of the paths costs. In D-SO there may be cooperation among individuals, or a centralized system may coordinate the route assignment. With the incorporation of *stochastic* approaches to solve this problem, new equilibrium criteria are introduced, the s*tochastic user equilibrium* (S-UE) and the s*tochastic system Optimum Assignment* (D-SO). In Fig. 1, this classification can be seen.

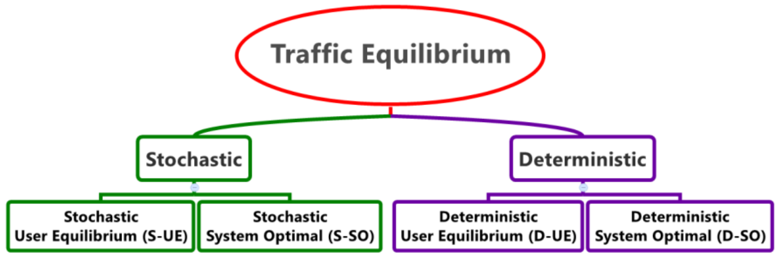


Fig. 1. Traffic Equilibrium Classification

The optimization of transportation on a road network is commonly formulated by means of two different mathematical formulations, the *static traffic assignment* (STA), and the *dynamic traffic assignment* (DTA). The STA approach deals with networks in equilibrium where the traffic demand and the edge flows are constant over time. On the other hand, the DTA deals with the more realistic situation of time-dependant edge flows and congestion.

1.1.1 Static Traffic Assignment.

Static Traffic Assignment consists of assigning routes to a set of drivers with fixed *origins* and *destinations* under steady-state flow conditions. A model of the *equilibrium assignment* for both Wardrop's principles, the UE and the SO was first presented by Martin Beckmann, Bartlett McGuire and Christopher Winsten (BMW) [5]. This solution became a standard in transportation planning since it was introduced. Currently, as new approaches have emerged, solutions based on *dynamic traffic assignment* have become available for dealing with congestion and variable conditions.

1.1.2 Dynamic traffic assignment

Dynamic Traffic Assignment refers to a broad variety of problems that deals with time-varying flows and originated with the seminal works of Yagar in 1971 [6], [7], and Merchant and Nemhauser in 1978 [8], [51]. These models aim at representing in time harmony the interaction between route choices, traffic flows, and cost metrics in [14]. DTA problems can be classified as [8]: *Mathematical programming formulations, Optimal control formulations, variational inequalities formulations*, and *Simulation-based models* [8]. The first three types of approaches suffer from problems induced by the inability of these algorithms to model human behavior. These may result in artificial delays at junctions in order to optimize traffic flow [8], [15], and first in, first out (FIFO) violations, where vehicles are forced to overtake other vehicles that departed earlier in order to optimize the traffic flow [8], [16], [15]. These problems are not consistent with traffic realism [8]. *Simulation-based models* utilize *traffic flow propagation models* to model the critical constraints that regulate traffic flow even though the problem abstraction could be analytical. The simulation is used as an approach to produce realistic solutions that satisfy the FIFO constraint and prevent the artificial delays. Among the *simulation-based models*, computational intelligence approaches have gained popularity recently. We briefly describe them in the next section.

1.1.3 Computational Intelligence Approaches to the DTA.

Much effort has been dedicated to these *stochastic* approaches based on nature-inspired methodologies. A new equilibrium criterion is needed in this area, the s*tochastic user equilibrium* (S-UE). For the traffic flow from a given *origin* to a certain *destination* by means of a number of available routes, in S-UE no user believes he can improve his journey time by unilaterally changing routes [9]. Achieving UE in this type of method is in general a very difficult task unless some simplifications are made, like in the *Stochastic Network Loading* (S-N-L) type of problems, which assumes that the measured travel times are independent from the edge flows [9]. Social insects exhibit efficient behaviors for path selection [20], and recent computational intelligence approaches based on *ant inspired* algorithms do not require these assumptions [12], [18], [38], [39], [40], [41], [42], [43], [52].

1.2 The Traffic Assignment Problem in VANETs

For a long time the predominant methods to assign traffic have focused on static models. There is a need for DTA algorithms for VANETs, but, at the same time, these dynamic models present unique complexities. The solutions to the DTA problem, at any given time, may include the cars on the road as well as those which will enter the road network in the future, or probably reschedule the departure time of the system's users [14]. The inclusion of the cars that will enter the road could be approximated considering past experience but would be inaccurate and alone would not solve the case of special events or accidents. To compensate for this, some algorithms may include feedback from the system to reroute some of the vehicles. Rescheduling users' departure time may be impractical because most of the traffic flow is due to activities with fixed schedules, like work or school.

As mentioned in Section 1.1.2, realistic solutions to *traffic assignment* may be compromised by the analytical models and algorithms, which introduce artificial delays in junctions and FIFO violation. The use of *Simulation-based models* is justified as these models closely represent the critical constraints that regulate traffic flow.

1.3 Motivation

VANET technologies will enable real-life traffic assignment in the near future. Traffic flow optimization has been studied since 1850 [53] and used to plan and design road networks, but a real-time system that takes vehicle interactions in real time has never been implemented. Incorporating wireless technology and information systems for traffic control purposes has been an area of research since the 1970s [54]. In this period, a large effort was dedicated to The Comprehensive Automobile Traffic Control System (CACS) by the Japanese government. Nowadays, communication standards have been adopted and the cost of technology has dropped, making feasible the implementation of VANET systems [55]. Nevertheless, VANETs systems are still under development and traffic assignment algorithms are still in their infancy.

There is a disproportionate increase of the vehicle miles travelled, which has doubled in recent years, compared to a modest 5% increase of the highway lane miles added. *Traffic assignment* aims at distributing routes to vehicles to improve travel time. *Traffic assignment* is a relatively new and active field of research that involves human behavior that is difficult to model and approaches that sometimes produce unrealistic solutions.

For many years the only methods available were intended for the STA and used to predict the dynamic road traffic. It was not until the 1970s that DTA started to be studied formally. Different analytical approaches have been proposed, but the mathematical solutions they provide usually suffer from deficiencies introduced by the methods themselves where unrealistic solutions are generated by the algorithms when minimizing travel costs.

On the other hand, approaches based on simulations are of interest as they are able to better model the constraints that lead to realistic solutions. Some of the simulation approaches have been based on ACO, which is also a relatively new field of research. However, most of these approaches have been intended for the static traffic assignment and, therefore, they would not provide accurate results in the presence of unanticipated conditions, such as accidents or short term road works. Other ACO approaches are based on the existence of infrastructure support, which limits the application of the algorithm to cases where a costly infrastructure is already in place.

Common difficulties that algorithms solving traffic assignment in VANETs face include the broadcast storm that happens when new messages are broadcasted and many vehicles attempt to retransmit the message. This situation results in unnecessary use of the communication channels. Another issue found in these networks is fragmentation, even in the presence of infrastructure. Many algorithms base their operation on the complete knowledge of the traffic demand, a situation that is difficult to implement in the real world. Any algorithm attempting to solve DTA in VANETs should be able to operate in the presence of partial information.

There is a need for algorithms for the dynamic traffic assignment problem that produces realistic solutions. Algorithms of this kind should avoid the broadcast storm and should be able to operate under complete or partial knowledge of the road network conditions and be able to operate with or without infrastructure support.

1.4 Problem Statement

We aim at generating an online DTA algorithm in VANETs. Traffic assignment is a very important problem as miles traveled increase faster than lane miles and intelligent wirelessly connected systems appear to be the immediate solution, if not the only one. Just in the U.S., from 1985 to 2006 the Vehicle Miles Traveled has doubled while the lane miles have just increased from 8 to 8.4 million [1]. The need for incorporating wireless technology and information systems for traffic control has been recognized since the 1970s [54], when a large effort was dedicated to improving road traffic by the Japanese government and the need for a two-way communication link was recognized as inevitable. Nowadays, communication standards have been adopted and the cost of technology has dropped, making the implementation of VANET systems feasible [55]. Finding solutions to the traffic assignment problem in VANET environment involves unique challenges as these networks face big challenges including [56]:

1. Rapid changes in topology,
2. Frequent fragmentation, even in the presence of supporting infrastructure,
3. Small effective network diameter (Diameter for which 90% of the included nodes are connected),
4. Potential large scale,
5. Variable network density, and
6. Topology affected the by drivers' behavior.

Existing DTA algorithms do not meet this challenges adequately. Analytical algorithms present FIFO violations and artificial delays at intersections. Algorithms based on simulations are very important since they are able to produce realistic solutions. An important group of simulated algorithms, *ant inspired* algorithms, are intended to obtain minimum cost paths. They have been used for *traffic assignment*. All these methods have mostly been intended for the STA or used in the centralized approach. Centralized algorithms for VANETs require the existence of an infrastructure that may not always be available, suffers from great computational complexity, and may still be subject to fragmentation. In conclusion, a desirable property of the needed DTA algorithms is that they should be distributed. Lastly, very few DTA algorithms exist specifically designed for VANETs due to the additional challenges imposed by this technology.

1.5 Contributions

We introduce a reactive ant inspired algorithm that works in presence of partial road network knowledge with or without infrastructure support. The algorithm is inspired by ACO in which scent marks (pheromones) are used to signal the quality of paths and evaporation to control the persistence of these marks. We summarize our contributions as:

1. We contribute a chapter on Swarm Intelligence-Inspired Routing Algorithms for Ad Hoc Wireless Networks [20]. In this chapter Swarm-intelligence inspired routing algorithms are introduced as an alternative for classical routing algorithms in ad hoc wireless networks. We present the biological principles that inspire these algorithms, and introduce some important swarm intelligence algorithms.

2. We contribute extensions to trafficmodeler to allow importing Open Street Maps (OSM) and make rapid VANET simulations [57]. Trafficmodeler [58] is a program intended for modeling traffic demand which is included in the Simulation of Urban MObility (SUMO) distribution [59]. OSM is a collaborative project to create a free editable maps of the world [60].

3. We contribute extensions to the Open Source vehicular network simulation framework Veins [61] to allow for traffic rerouting on VANET simulations (W. Arellano, I. Mahgoub, and M. Ilyas, "Veins extensions to implement a message based algorithm for Dynamic Traffic Assignment in VANETs simulations." In High-capacity Optical Networks and Emerging/Enabling Technologies (HONET), 2014 11th Annual, pp. 29-35. IEEE, 2014) [62].
4. We contribute a DTA algorithm inspired by ants behavior that minimizes the broadcast storm as it uses speed thresholds to initiate the traffic report activities and uses a system of on the fly clusters formation of limited duration. In this system, cluster heads are in charge of the information gathering and message broadcasting [63]. This is the first ant algorithm that uses pheromones to mark the bad paths.

We continue in the next section with the organization of this document.

1.6 Organization

The rest of this document is organized as follows: In Section 2 we present a traffic assignment overview. In Section 3 we introduce ant colony optimization. Section 4 introduces aggregation. In Section 5 we present a literature survey. Section 6 introduces our algorithm. In Section 7 we extend the algorithm for larger networks. Finally, in Section 8 we present our conclusions and indicate future work to continue this project.

2. TRAFFIC ASSIGNMENT OVERVIEW

When optimizing the flow of vehicles on road networks we need to consider models to represent these networks and specific kinds of mathematical problems. When we optimize traffic flow in road networks we are dealing with *transportation planning models* and the important associated *traffic assignment* problems. *Traffic assignment* is an active field of research and new approaches are frequently found. In the next sections we briefly introduce *transportation planning models* that are commonly used in this field, the associated *traffic assignment* problems, the *traffic flow models* usually considered to test the quality of the proposed solutions, and, in particular, the *microscopic simulation models* which are the preferred approach when working with VANETs.

2.1 Transportation Planning Models

The need to travel of households in transportation systems arises from decisions about their needs to participate in spatially separated social, economical, and cultural activities. The ensemble of these activities is called the *activity system* [13]. Even though there are *transportation planning models* that can be used to model the growth and layout of urban areas, we are centered in the day to day traffic on well-established road networks. There are two widely used approaches for this purpose, *Trip-based transportation planning models* and *Activity-Based Transportation Models.* Even though both models start from a population activity knowledge, in the first case, when the *trips* are generated, the links to activity, behavior, and time are lost [64] and limitations are imposed on the model [65]. We now proceed to explain the main models in use.

2.1.1 Trip-based transportation planning models.

The application of this model is nearly universal [64]. Travel has usually been modeled as *origin-destination* (OD) trips even though, in theory, it has always been considered as derived from the demand for activity participation [64]. The link between travel and activities was first introduced in [66], where the authors developed a *four step model* that was later institutionalized in the Federal-Aid Highway Act of 1962 [67]. The next four sections outline the *four step model*.

2.1.1.1 Trip Generation. In this step trip ends are separated into productions and attractions generated at the zone, household, or individual level for different trip purposes. Usually, three or more different trip purposes are used with popular home-based work trips (HBW), home-based other trips (HBO), and non-home-based trips (NHB). The trip productions and attractions are computed using regression analysis or similar techniques from the absolute counts of trips departing and arriving at each zone and the system's activity characteristics, translating an *activity-based* system description into a *trip-based* one. Productions generated at the household level and attractions at the zone level are used frequently, time dependence can be reintroduced by computing productions and attractions for specific periods of time [64].

2.1.1.2 Trip distribution. The objective of this step is to link all trip origins (productions) to destinations (attractions) and obtain a complete OD table or matrix. In this matrix, element od_{ij} corresponding to the intersection or row i and column j

represents the number of trips from origin O_i and destination D_j [13]. To obtain OD some network attributes, typically, inter-zonal travel times are required [64]. In general, the process of obtaining the OD is underdetermined as the number *origin-destinations* $c = n(n-1)$ where n is the number of zones but the number of links is usually $O(n)$ [68]. This implies that an infinite number of solutions may exist [69]. Probably the most popular method to compute OD is the *production-constrained gravity-model* [64] in which the association of origin destinations are obtained from *productions*, *attractions,* and an *impedance function* that can be estimated from *empirical* or *intuitive* methods with *friction factors* being the most popular. *Friction factors* are exponential or gamma functions obtained from travel frequencies from household surveys.

2.1.1.3 Mode Choice. After the *origin-destination* matrix is obtained, the next step consists of selecting the different *modes of transportation* to be used by the users of the transportation system. Examples of *transportation modes* include: private, public, vehicular, railroad, etc [13]. In other words, the OD is decomposed according to the different *transportation modes.* To accomplish this process *discrete choice theory* is a frequent choice. *Mode choice* could be combine with *trip distribution* but, in this case, an adjusted gravity model and an impedance function are used [13].

2.1.1.4 Traffic assignment. The final step of the *four step model* is the traffic assignment. In this context, routes are defined as a sequence of links or roads between an *origin* and a *destination* and the trips are also known as *travel* or *traffic demand. Traffic assignment* is the process of assigning the *traffic demand* to the routes. Some implementations of the *four step method* include iterations and feedback from this stage to steps *b)* and *c)*. Interestingly the UE equilibrium has been related to *Nash Equilibrium* in *Non-Cooperative games* [70], [17].

2.1.2 Activity-Based Transportation Models.

In the *Trip-based transportation planning models*, trips are generated independent of individual activities and their associated relationship with space and time and without choice alternatives [65]. This obviously imposes serious limitations to how accurate these models represent traffic demand. Independently of these limitations, the first ideas on *Activity-Based Transportation Models* originated in the 1970s [65] and were first introduced as a comprehensive study in 1983 [71]. The *Activity-Based Transportation Models* do not have an explicit step-by-step procedure similar to the *four step model*. However, some ingredients are now recognized [13], [72]:

2.1.2.1 Generation of Activities. This process has required the replacement of the traditional *travel diary surveys* with *activity-based surveys.* There is a need for precise definitions of terms, as proposed in [72], for *journey* as a tour starting and ending at the relevant location for a person, with *tour* sequence of trips starting and ending at the same location, where *trips* are continuous sequences of stages between two activities, and *stages* are vehicle motions including the idle time during and immediately before the stage. Reference [72] also proposes characterizing activities by *kind, purpose* (what the person hopes to achieve), *moral meaning, project* (greater context), *duration, effort, expenditure/income,* and *urgency.*

2.1.2.2 Modeling of Household Choices. *Activity-based models* consider the household members interactions and their impact on their travel behavior. The purpose is to model whole daily activity chains. Destinations and mode choices are combined with low-level decisions, such as parking choices or high-level decisions such as car ownership and place of residence [72].

2.1.2.3 Scheduling of Activities. Nowadays, the Scheduling of Activities includes elements for *long-term commitments* of the household and its members to their life style and to each other, *medium-term calendar* of each person including household tasks assigned to them and their personal activity demands, and a *daily calendar* with the discrete scheduled events for each person. The scheduling process works from the *daily calendar* but is able to interchange events with the *medium-term calendar* and produce changes to the *long-term commitments* on a one-off basis. The scheduling process provides activity formulation for emotional and physiological needs and ultimately would include *day-to-day learning.* [72].

2.2 Traffic Assignment Problems

As we mentioned before, the optimization of transportation on a road network is commonly formulated by means of two different mathematical formulations: the *static traffic assignment*, and the *dynamic traffic assignment*. DTA deals with the more realistic situation of time dependant link flows and congestion. Ignoring these time dependencies may originate inconsistencies like the Smeed's paradox, in which vehicles departing on high-flow conditions can arrive later than vehicles that departed later on a low-flow condition [73].

2.2.1 The Static Traffic Assignment.

The *Static Traffic Assignment*, consists of assigning routes to a set of drivers with fixed *origins* and *destinations* under steady-state flow conditions. Reference [13] indicates that different approaches can be used to solve *the static traffic assignment:*

1. An *all-or-nothing assignment* (AON) in which all drivers choose the same cheapest route is based on all drivers having perfect knowledge of the link's impedances and that these are constant.

2. A *stochastic assignment* is possible when drivers have imperfect knowledge of the constant link's impedances and these are chosen using probability distribution functions after which an AON is performed. The method can be iterated until certain criterion is met.

3. An *equilibrium assignment* incorporates the concept that an increase in the traffic flow on the links changes the link impedances, which may change the cheapest routes.

4. A *stochastic equilibrium assignment* is also possible similar to as explained in 2. when the drivers have imperfect knowledge of the road network status.

As mentioned before, a model of the *equilibrium assignment* for both Wardrop's principles, the UE and the SO was first presented by BMW [5]. In [74] this approach is compared with the more recent model presented by Nesterov and De Palma (NdP) [75]. Following the descriptions in [74], [76], [77], in the next three sections we will proceed to formally define the *equilibrium assignment* problem and briefly describe the models of BMW and NdP.

2.2.1.1 Nonlinear Programming. Many important contributions to the STA problem have been done in the context of the *nonlinear programming problem* (NLP). In general, an optimization problem deals with minimizing or maximizing a real objective function $f(x_1,\ x_2,\ \cdot\cdot\cdot\cdot\cdot\cdot\cdot\ ,\ x_n)$ by finding the values of n real variables $x_1,\ x_2,\cdot\cdot\cdot\cdot\cdot, x_n$ within a certain feasible region [78]:

Maximize or *minimize* $\quad f(x_1,\ x_2,\cdot\cdot\cdot\cdot\cdot\cdot, x_n),\ subject\ to\ (s.t.)$:

$$g_i(x_1,\ x_2,\cdot\cdot\cdot\cdot\cdot\cdot, x_n) \leq b_i \text{ for } i = 1,\cdot\cdot\cdot\cdot\cdot\cdot, m \tag{1}$$

The problem is called a *nonlinear programming problem* (NLP) if the objective function f and/or any of the g_i functions that define the feasible region are nonlinear. Within the general case of the *nonlinear programming problem*, if the objective function f is convex or concave and all the g_i functions are convex, the problem is called *convex* and any local maximum or minimum of f must be global and, in general, optimization is easier. One important case of NLP is that where:

Maximize or *minimize* $\quad f(\boldsymbol{x}),\ \boldsymbol{x} = x_1,\ x_2,\cdot\cdot\cdot\cdot\cdot\cdot, x_n,\ subject\ to\ (s.t.)$:

$$g_i(\boldsymbol{x}) = b_i \text{ for } i = 1,\cdot\cdot\cdot\cdot\cdot\cdot, m$$

In this case, expressions can be found for the maxima or minima by means of Lagrange's theorem. We define the Lagrange function $\mathcal{L} = f(\boldsymbol{x}) + \sum_{j=1}^{m}\left(\lambda_j g_j(\boldsymbol{x}) - b_i\right)$ where coefficients λ_j are the Lagrange multipliers. The theorem states that the solution to the original problem can be obtained by solving the system of equations:

$$\frac{\partial}{\partial x_i}\mathcal{L} = 0,\ i = 1,\cdots\cdots,n$$

$$g_i(\boldsymbol{x}) = b_i \text{ for } i = 1,\cdots\cdots,m.$$

By means of Lagrange's theorem the problem was transformed from a constrained optimization into a system of $n + m$ equations, which usually is simpler to solve.

A generalization of Lagrange's multipliers was introduced by Kuhn and Tucker [79]. They presented some necessary conditions for a feasible solution $\boldsymbol{x}$ of an *inequality constrained optimization problem* to be optimal. Now these conditions are also known as the Karush–Kuhn–Tucker (KKT) conditions because it was later discovered that Karush proved these conditions earlier in a different context [80]. The Karush–Kuhn–Tucker conditions can be stated as:

Maximize or *minimize* $\quad f(x_1,\ x_2,\cdots\cdots,x_n),\ s.t.$:

$$g_i(x_1,\ x_2,\cdots\cdots,x_n) \geq 0\ for\ i = 1,\cdots\cdots,m \tag{2}$$

$$\mathcal{L} = f(\boldsymbol{x}) + \textstyle\sum_{j=1}^{m}\big(\lambda_j g_j(\boldsymbol{x}) - b_i\big)$$

$$\frac{\partial}{\partial x_i}(\mathcal{L}) \leq 0, \quad x_i \geq 0, \quad x_i\frac{\partial}{\partial x_i}(\mathcal{L}) = 0 \quad for\ \ i = 1,\cdots\cdots,n \tag{3}$$

$$\lambda_j \geq 0, \quad \lambda_j g_j(\boldsymbol{x}) = 0, \quad for\ \ j = 1,\cdots\cdots,m \tag{4}$$

Note that $g_i(x_1,\ x_2,\cdots\cdots,x_n) \geq 0$ for $i = 1,\cdots\cdots,m$ can be obtained from (1) if the constant b_i is absorbed by g_i and the inequality is rearranged (the mathematical expressions for g_i for each case are different). To conclude this section, we would like to mention that in the BMW model [5], the optimality of the UE was guaranteed thanks to the existence of the Karush–Kuhn–Tucker conditions [80]. We will present BMW's model in Section 2.2.1.3.

2.2.1.2 Equilibrium Assignment. The goal of the *equilibrium assignment* is to allocate routes on the road network to a set of drivers with fixed *origins* and *destinations*, in order to obtain either UE or SO states. We will start defining some nomenclature. We represent the road network with a graph $\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{A})$ where vector $\boldsymbol{N}$ represents the nodes (intersections) and vector $\boldsymbol{A}$ represents the arcs (roads). For each arc $a \in \boldsymbol{A}$ we define the *capacity*, c_a as the maximum number of cars that that can cross a in a given period of time and the *free travel time*, $\bar{t}_a$ as the minimum travel time required when traversing a at the maximal allowed speed. We define the capacity vector and the free travel time as:

$$\boldsymbol{c} \coloneqq (c_1, c_1, \cdot \cdots \cdot, c_{|A|}) \in \mathbb{R}^{|A|}$$

$$\bar{t}_a \coloneqq (\bar{t}_1, \bar{t}_2, \cdot \cdots \cdot, \bar{t}_{|\mathrm{a}|}) \in \mathbb{R}^{|A|} \quad \forall a_i \in \boldsymbol{A}$$

The goal of the *equilibrium assignment* is to allocate routes on the road network to a set of drivers with fixed *origins* and *destinations*, in order to obtain either UE or SO states. The current state of the network is represented by a *flow* vector, $\boldsymbol{f}$, representing the number of vehicles entering the arcs per unit of time, and a *travel time* vector, $\boldsymbol{t}$, representing how long it takes traverse the arcs:

$$\boldsymbol{f} \coloneqq (f_1, f_1, \cdot \cdots \cdot, f_{|A|}) \in \mathbb{R}^{|A|}$$

$$\boldsymbol{t} \coloneqq (t_1, t_1, \cdot \cdots \cdot, t_{|A|}) \in \mathbb{R}^{|A|}$$

We represent with $\boldsymbol{OD} \subset \boldsymbol{N} \times \boldsymbol{N}$ the set of fixed *origins* and *destinations* of the road network. For each *origin destination* pair $\boldsymbol{k} \in \boldsymbol{OD}$, $d_k > 0$ represents the fixed trip rate of drivers per unit of time traveling from the *origin* of $\boldsymbol{k}$ to its destination. Let us denote with $\boldsymbol{R_k}$, the set of all routes or paths that connect the *origin* of $\boldsymbol{k}$ with its *destination*. A route $\boldsymbol{r} \in \boldsymbol{R_k}$ is a vector $\boldsymbol{r} \in \mathbb{R}^{|\boldsymbol{A}|}$ with coordinates equal to 0 for each arc that does not belong to the route or to an integer indicating the number of times each arc is used otherwise (values greater than 1 indicate the presence of loops). Vector $\boldsymbol{f^k} \in \mathbb{R}^{|\boldsymbol{A}|}$, represents the *flow* on pair $\boldsymbol{k}$**,** vector $\boldsymbol{f_r^k} \in \mathbb{R}^{|A|}$ represents the flow on route $\boldsymbol{r}$ of $\boldsymbol{k}$, and finally, vector $\boldsymbol{f_a^k} \in \mathbb{R}^{|A|}$, represents the flow of $\boldsymbol{k}$ on arc a. In other words, vectors $\boldsymbol{f^k}$**,** $\boldsymbol{f_r^k}$**,** and $\boldsymbol{f_a^k}$ have as many components as arcs present in the road network. The values of the components of $\boldsymbol{f_r^k}$ are either the constant flow value d_k^r, on route r, if the corresponding arc belongs to r, or 0 otherwise. The values of the components of $\boldsymbol{f_a^k}$ are all 0 except for the component corresponding to arc a, which equals the flow in that arc due to $\boldsymbol{k}$. If we denote by $\boldsymbol{f_a}$ the total flow on arc a, we can express the road network flow as:

$$\boldsymbol{f} = \sum_{\boldsymbol{k} \in \boldsymbol{OD}} \boldsymbol{f^k} = \sum_{a \in A} \boldsymbol{f_a} \tag{5}$$

Where $\boldsymbol{f^k}$ can be obtained:

$$\boldsymbol{f^k} = \sum_{r \in \boldsymbol{R_k}} \boldsymbol{f_r^k} \tag{6}$$

with $\boldsymbol{f_a^k}$ expressed as:

$$\boldsymbol{f_a^k} = \sum_{\{r \in \boldsymbol{R_k}, \boldsymbol{a} \in \boldsymbol{r}\}} \boldsymbol{1_a} \left(\boldsymbol{f_r^k} \circ \boldsymbol{1_a} \right) = \sum_{\{r \in \boldsymbol{R_k}, \boldsymbol{a} \in \boldsymbol{r}\}} \boldsymbol{1_a} d_k^r \tag{7}$$

where $\mathbf{1}_{\boldsymbol{a}} \coloneqq (\nu_1, \nu_2, \cdots\cdots, \nu_{|A|}) \in \mathbb{R}^{|\boldsymbol{A}|}$ where $\nu_i = 1$ if $i = a$ and $\nu_i = 0$ otherwise. Finally,

$$\boldsymbol{f_a} = \sum_{\boldsymbol{k} \in \boldsymbol{OD}} \boldsymbol{f_a^k} = \sum_{\boldsymbol{k} \in \boldsymbol{OD}} \sum_{\{r \in \boldsymbol{R_k}, \boldsymbol{a} \in \boldsymbol{r}\}} \mathbf{1}_{\boldsymbol{a}} d_k^r \quad \forall a \in A \tag{8}$$

We can now define a feasible s*tatic traffic assignment* as a solution $(\boldsymbol{f}, \boldsymbol{t})$ that allocates the drivers on the routes constrained to their fixed flows from their *origins* to their *destinations* and the flows in any route being positive. We will conclude this section by summarizing the models of BMW and NdP. BMW's model was the first model to represent the s*tatic traffic assignment* under SO and UE equilibriums, and it has been predominant for the STA but has been recently challenged by the model of NdP.

2.2.1.3 Beckmann, McGuire and Winsten Model. In this model it is assumed that the travel time on arc a, t_a, depends only on $\boldsymbol{f_a}$ and that it is defined by a continuous, convex, nonnegative, non-decreasing latency function $l_a(\circ)$. The cost of traveling on an arc a is defined as $f_a l_a(f_a)$. To take into consideration the capacity limit c_a, function $l_a(\circ)$ is chosen in a way that penalizes capacity violations (still this may permit solutions where the capacities are violated). The *static traffic assignment* is the solution $\left(\boldsymbol{f}, \boldsymbol{l}(\boldsymbol{f})\right)$ where $\boldsymbol{l} \coloneqq \left(l_{a_1}, l_{a_1}, \cdots\cdots, l_{a_{|A|}}\right) \in \mathbb{R}^{|A|}$ $\forall a_i \in \boldsymbol{A}$ is optimized to produce either SO or UE.

1. *SO optimization*. The convex problem of optimizing the total travel time $\sum_{a \in A} f_a l_a$, is solved:

$$min_f \ \sum_{a \in A} f_a l_a (f_a)$$

$$\sum_{\boldsymbol{r} \in \boldsymbol{R_k}} d_k^r = d_k \tag{9}$$

$$f_r^k > 0 \ \ \forall k \in OD$$

$$\boldsymbol{f_a} = \sum_{\boldsymbol{k} \in \boldsymbol{OD}} \boldsymbol{f_a^k} \ = \sum_{\boldsymbol{k} \in \boldsymbol{OD}} \sum_{\{r \in \boldsymbol{R_k}, \boldsymbol{a} \in \boldsymbol{r}\}} \boldsymbol{1_a} d_k^r \ \ \forall a \in A$$

Where d_k^r represents the flow of vehicles from origin destination pair $\boldsymbol{k}$ traveling on route $\boldsymbol{r}$. Usually the Bureau of Public Road (BPR) function for the latency is used [81]:

$$l_a(f_a) = t_a \left(1 + \alpha \left(\frac{f_a}{c_a}\right)^{\beta}\right) \tag{10}$$

In this function, parameters α and β determine the penalty for overflowing the arc capacity. The original values provided in this publication were $\alpha = .15$ and $\beta = 4$.

1. *UE optimization.* The convex problem of optimizing the travel time of the individual vehicles is solved. We define $l_r(f) = \sum_{a \in \boldsymbol{r}} l_a(f_a)$ as the travel time of vehicles on route $\boldsymbol{r} \in \boldsymbol{k}$. The problem consists of finding $\boldsymbol{r}$ such that $l_r < l_q \ \ \forall r.q \in \boldsymbol{R_k}$. It can be shown that this is equivalent to optimizing the following convex problem:

$$min_f \ \ \sum_{a \in A} \int_0^{f_a} l_a(x) dx$$

$$\sum_{\boldsymbol{r} \in \boldsymbol{R_k}} d_k^r = d_k \tag{11}$$

$$f_r^k > 0 \ \ \forall k \in OD$$

$$\boldsymbol{f_a} = \sum_{\boldsymbol{k}\in\boldsymbol{OD}} \boldsymbol{f_a^k} \; = \sum_{\boldsymbol{k}\in\boldsymbol{OD}} \sum_{\{r\in\boldsymbol{R_k},\boldsymbol{a}\in\boldsymbol{r}\}} \mathbf{1}_{\boldsymbol{a}} d_k^r \quad \forall a \in A$$

The last constraint is a link route formulation of the problem that BMW originally presented as link node representation with an independent formulation of the node conservation of flow similar to Kirchhoff's current law [80]. It was within this representation that BMW formulated the conditions that would guarantee a unique optimal solution based on the KKT conditions. In the original notation, $x_{ij,q}$ represents the flow from node i to node q. The flow going to node q from node i equals the flow from node q to node i plus the flow originating in node i to node q:

$$\sum_j x_{ij,q} = \sum_j x_{ji,q} + x_{i,q}$$

Under UE drivers select the lowest cost route from node i to node q with cost $y_{i,q}$. It holds that if drivers take an alternate route that includes adjacent node *j* the cost will satisfy:

$$y_{i,q} \leq y_{ij} + y_{j,q}$$

where y_{ij} is the cost of traveling in arc (i, j). Under the assumption of UE, if $x_{ij,q} > 0$ then $y_{i,q} = y_{ij} + y_{j,q}$ and, if $x_{ij,q} = 0$ then $y_{i,q} < y_{ij} + y_{j,q}$. This means that under this equilibrium, drivers would never take a route with higher cost than an existing alternative, and we can conclude that, under UE, the constraint $x_{ij,q}\left(y_{i,q} - y_{ij} - y_{j,q}\right) = 0$ always holds. When applying KKT to the UE problem of maximizing $-\frac{1}{2}\sum_{ij} h_{ij}\left(x_{ij}\right)x_{ij}$ (the optimization consisted on maximizing a negative cost), where $y_{ij} = h_{ij}\left(x_{ij}\right)$ is the capacity function, BMW found that the interpretation of the KKT conditions involved

$h'_{ij}(x_{ij})$ in the necessary conditions of the optimal solution. They simplified the problem by introducing an equivalent artificial function that would cancel $h'_{ij}(x_{ij})$ [80]:

$$max_{x_{ij,q} \geq 0} \quad -\frac{1}{2}\int_0^{x_{ij}} h_{ij}(x)\, dx$$

The $\frac{1}{2}$ is used in both cases because of the way flow was defined including traffic in both directions. Essentially, with some style changes, this is the SO *optimization* that we presented at the beginning of this section.

2.2.1.4 Nesterov and De Palma Model. This new model tries to address some problems that may arise in the BMW model [82]. The BMW model assumes that the cost of traveling on an arc a is $f_a l_a(f_a)$ with $l_a(\circ)$ a non-decreasing function. This may lead to a number of contradictions such as:

1. The formulation with function $l_a(\circ)$ allows for solutions that exceed the capacity of the road.
2. The contradiction that a small f_a can arise from two different conditions, low traffic or congested road

To address the problems mentioned above, the model presented by NdP basically changes two concepts of the BMW model: First, there are no congestions and the travel time on any arc a equals the *free travel time*, $t_a = \bar{t}_a$. Second, the cost of traveling on arc a is equal to $f_a t_a$. The SO formulation for this model is very similar to that of the BMW model, but with the new cost formulation: $\sum_{k \in OD} f_a \bar{t}_a$. The UE equilibrium is more complicated as the existence of the solution is not directly guaranteed as with the BMW model but needs to be determined with the Lagrange multipliers.

To conclude this section, we note the extraordinary contribution of Martin Beckmann, Bartlett McGuire and Christopher Winsten, who first modeled the STA setting a model that has been the standard up to today. Nesterov and De Palma presented some valid criticism and formulation that may contribute to future development. We cannot conclude this section without mentioning the algorithm of Marguerite Frank and Philip Wolfe, the Frank-Wolf algorithm (FW) [83]. Le Blanc et.al [84] utilized the algorithm in a small city network after which the method became a standard for 30 years [85]. The algorithm suffers from low convergence but is highly efficient in memory usage. This is a typical algorithm using link-based aggregation over the *origin destination* pairs, these algorithms convey the highest level of aggregation and solve the traffic assignment using the total traffic flow for each link. Other methods have been proposed using route-based approaches, which are less memory efficient but present high accuracy in reasonable amounts of time. Fewer approaches have been developed on an *origin-based* approach; of these the Bar-Gera Algorithm has represented a breakthrough of great importance in the field [85]. The Bar-Gera algorithm offers highly accurate solutions with reasonable use of time and memory. The key point of the Bar-Gera algorithm is its capacity to disaggregate cyclic traffic flows which results in an algorithm with complexity that grows linearly with the network size and provides the same level of detail as route aggregation. Currently, as new approaches have emerged, solutions based on *dynamic traffic assignment* have become available for dealing with congestion and variable conditions. In the next section we will proceed to present the most important types of these algorithms.

2.2.2 The Dynamic Traffic Assignment.

The *dynamic traffic assignment* refers to a broad variety of problems dealing with time-varying flows that originated with the seminal works of Yagar in 1971 [6], [7], and Merchant and Nemhauser in 1978 [51], [8]. These models aim at representing the interaction between route choices, traffic flows, and time and cost metrics in coherent fashion with time flow [14]. A literature review on the DTA is presented in [8]. This document indicates that in the general DTA problems, if traffic dynamics and driver behavior are modeled accurately, then the existence, uniqueness and stability of the solution cannot be guaranteed. The authors also propose a DTA taxonomy and state that, due to the complexities indicated earlier, great effort is dedicated to heuristic approaches. We proceed next to summarize the mentioned taxonomy.

2.2.2.1 Mathematical Programming Formulations, Merchant and Nemhauser

Mathematical programming formulations refer to optimization problems where a function $f(x_1, x_2, \cdot \cdot \cdot \cdot, x_n)$ is maximized or minimized constrained to the variables belonging to a subset of the domain of f, $\Omega = (x_1, x_2, \cdot \cdot \cdot \cdot, x_n) \subseteq \mathbb{R}^n$ [86]. The work of Merchant and Nemhauser falls under this category. They formulated a discrete-time nonlinear, non-convex programming approach for the SO case of multiple origins and a single destination. We proceed now to describe their model. The road network is represented with a graph $\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{A})$ where vector $\boldsymbol{N}$ represents the nodes (intersections) and vector $\boldsymbol{A}$ represents the arcs (roads). Time is divided into discrete intervals numbered $i = 0, 1, \cdot \cdot \cdot I$, the number of vehicles in a particular arc $j \in \boldsymbol{A}$ during time interval i is denoted by x_{ij}, and the cost of traveling on arc j at time interval i is $h(x_{ij})$. The optimization problem consists of:

$$minimize\ \sum_{i=1}^{I} \sum_{j=1}^{a} h_{ij}(x_{ij})$$

with the following constraints:

$$x_{i+1,j} = x_{ij} - g_j(x_{ij}) + d_{ij},\ i = 0,\ 1,\ \cdot,\ \cdot,\ \cdot,\ \cdot\ I-1,\ \forall\ j \in \boldsymbol{A}$$

$$\sum_{j \in A(q)} d_{ij} = F_i(q) + \sum_{j \in B(q)} g_j(x_{ij}),\quad i = 0,\ 1,\ \cdot,\ \cdot,\ \cdot,\ \cdot\ I-1,\ \forall\ q \in \boldsymbol{N}$$

$$x_{0j} = R_j \geq 0\ (given),\ \ \forall\ j \in \boldsymbol{A}$$

$$d_{ij} \geq 0,\ i = 0,\ 1,\ \cdot,\ \cdot,\ \cdot,\ \cdot\ I-1,\ \forall\ j \in A$$

$$x_{ij} \geq 0,\quad i = 0,\ 1,\ \cdot,\ \cdot,\ \cdot,\ \cdot\ I-1,\ \forall\ j \in A$$

where *j* and *q* represent generic arcs and nodes respectively. $g_j(x_{ij})$ represents the exit capacity of arc j. d_{ij} is the control variable that limits the access of vehicles to arc *j* at time interval i. $F_i(q)$ represents the external input to node q at time interval i. $A(q)$, $B(q)$ are the set of outgoing arcs and the set of incoming arcs at node q.

By means of a piecewise linear approximation of the model, the authors reduced the computational complexity and, by placing some restrictions on the cost function, they guarantee achieving a global minimum [51].

2.2.2.2 Optimal Control Formulations. The *discrete-time* mathematical program is replaced by a *continuous-time* optimal control formulation with constraints that are analogous to those for the mathematical programming formulations, but defined in continuous-time. The OD trip rates and link flows are represented as continuous functions of time. It is assumed that OD trip rates are known.

2.2.2.3 Variational Inequalities Formulations. *Variational Inequalities* (VI) is a recent mathematical field with application originally in partial differential equations mechanics. Those VI were *infinite dimensional* [87]. After the application by Dafermos to traffic equilibrium [88], the field was unveiled to applications in economics, management science, operations research, and engineering, specifically in transportation [87]. In [89] the *finite dimensional variational inequality problem* $VI(F, K)$ is defined as:

To find a vector $x^* \in K \subset \mathbb{R}^n$ such that

$$\langle F(x^*)^T, x - x^* \rangle \geq 0, \forall x \in K$$

where F is a continuous function from K to $\mathbb{R}^n$, K is a given closed convex set, and $\langle \cdot\,,\ \cdot \rangle$ is the inner product of an n-dimensional Euclidean space. A geometric interpretation of this definition indicates that $F(x^*)^T$ is orthogonal to the feasible set K at the point x^*. It can be demonstrated that the VI formulation allows for unified treatment of equilibrium and optimization problems [89], such as those encountered in *traffic assignment*.

2.2.2.4 Simulation-based models. This type of approach uses *traffic flow propagation models* to model the critical constraints that regulate traffic. The traffic simulation is used as an approach to produce realistic solutions that satisfy the FIFO constraint and prevent the artificial delays that affect the analytical formulations. We propose a classification for simulation-based approaches in the next section.

2.2.2.5 Classification of Simulation-Based Models. Suton [90] and Schmitt and Jula [91] present taxonomies for the DTA. In this section we present a taxonomy of simulation-based models traffic assignment algorithms. In Fig. 2 we introduce the classification for simulation-based models, which we proceed to explain next.

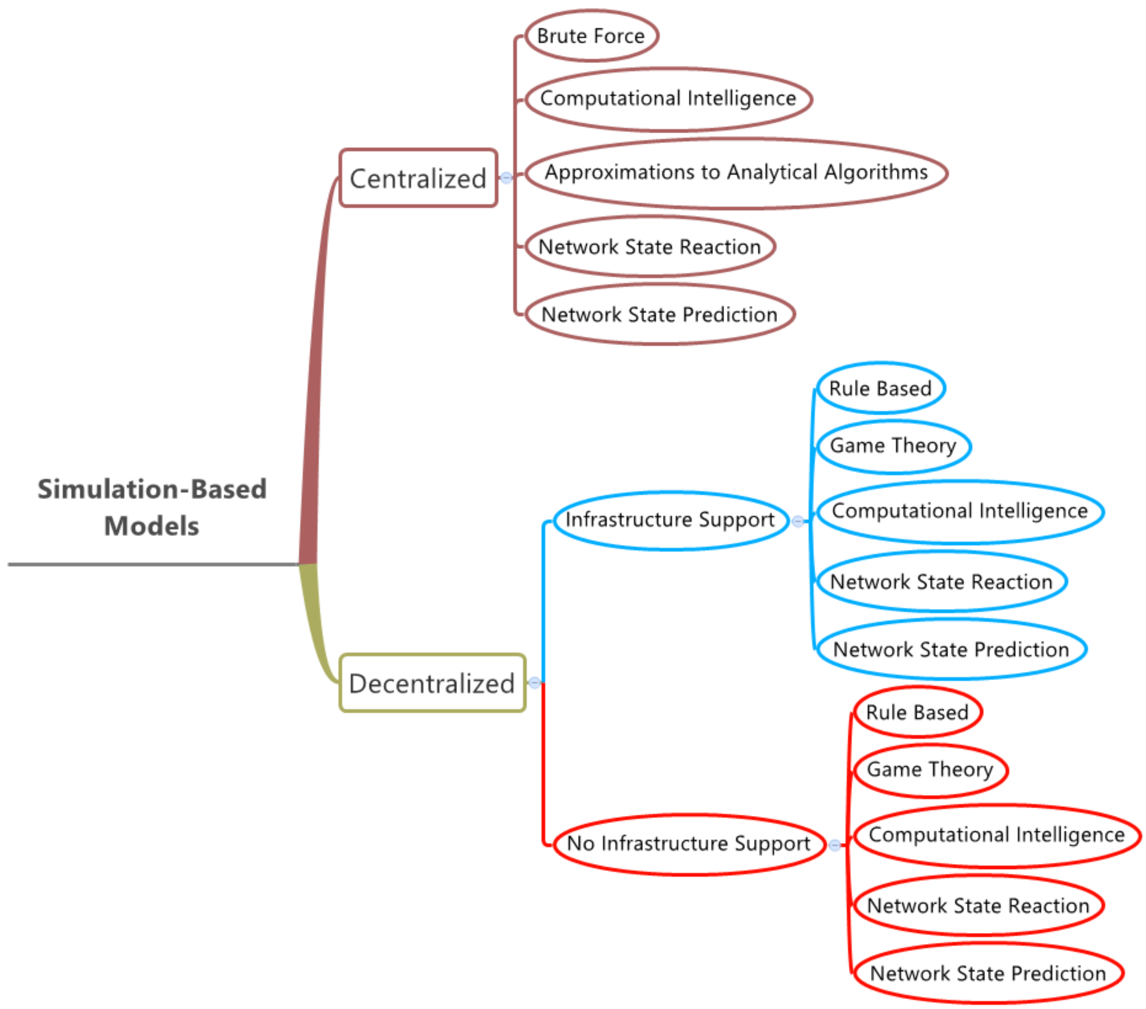


Fig. 2. Taxonomy of Simulation-Based DTA algorithms

2.2.2.5.1 **Rule Based Approaches.** In this type of simulation, vehicles possess a set of decision rules that can produce route changes according to the varying road conditions.

2.2.2.5.2 **Brute Force Approaches.** This type of algorithm runs the complete simulation in an iterative way, updating the link costs with the simulation results at each new iteration. These costs are used to produce traffic assignment.

2.2.2.5.3 **Game Theory Approaches.** This category groups algorithms inspired by game theory. Of particular interest are those based on Non-Cooperative Game Theory (NCGT).

2.2.2.5.4 **Computational Intelligence Approaches.** *Computational intelligence* attempts to solve complex problems by means of nature-inspired methodologies and the design of intelligent agents.

2.2.2.5.5 **Approximations to Analytical Algorithms.** Algorithms of this type use solutions by approximation to analytical formulations. Examples of this type include methods of successive approximations, such as the Method of Successive Averages (MSA) [92], [93].

2.2.2.5.6 **Network State Reaction Approaches.** Reactive route guidance algorithms are based solely on the current conditions of the network. The system aims at equal travel conditions for the routes in each OD for the current conditions.

2.2.2.5.7 **Network State Prediction Approaches.** This type of approach tries to anticipate the network state from history.

2.2.2.5.8 **Centralized Approaches.** A centralized system gathers information from the road network and makes the route selection for the vehicles.

2.2.2.5.9 **Decentralized Approaches.** The route choices are the result of interactions among the vehicles and the individual vehicles are the ones making those decisions.

2.2.2.5.10 **Approaches with Infrastructure Support.** In this type of approach, an infrastructure of devices along the road network is required to collect traffic information.

2.2.2.5.11 **Approaches with No Infrastructure Support.** In this type of approach, the traffic information is gathered by means of vehicle interactions only.

2.3 Traffic Flow Propagation Models

These models try to represent the physical propagation of traffic flows. They can be classified according to different criteria such as: scale of the independent variables (continuous, discrete, semi-discrete); level of detail (submicroscopic, microscopic, mesoscopic, macroscopic); representation of the processes (deterministic, stochastic); operationalization (analytical, simulation); scale of application (networks, stretches, links, and intersections) [94]. Of special interest is the level of detail classification, which we describe next.

2.3.1 Microscopic Simulation Models.

In these models the movement of each individual vehicle is described. The interaction of *vehicles characteristics* and *driver behaviors* are modeled in great level of detail in space and time. Chains of driver's decisions describe vehicle actions, such as lane changing. There is a large number of simulators of this kind; [95] identified 58 and analyzed 32 of them.

2.3.2 Submicroscopic Simulation Models.

These models are similar to *microscopic simulation* models, but, additionally, specific parts of the vehicles and the associated *control behavior* like changing gears are modeled in correspondence to the surrounding conditions [94].

2.3.3 Mesoscopic Simulation Models.

These models describe traffic flow at medium level of detail. The behavior rules, like acceleration and lane changing, are specified at individual level but in a more aggregated form (e.g. using probability distribution functions). The vehicles and drivers behavior are not traced at the individual level. Three popular mesoscopic simulation models are *headway distribution* models, *cluster models*, and the *gas-kinetic* continuum models.

2.3.4 Macroscopic Simulation Models.

In this type of simulators, individual vehicle maneuvers are not explicitly represented. Traffic is described at a high level of aggregation without distinguishing its individual constituents [94].

2.4 Microscopic Simulation Models

Many models have been developed to represent the behavior of individual vehicles in a road network traffic simulation. There are different approaches to Microscopic Simulation *Models.* Among them we can mention:

1. *Car following Models.* We present some of the most important models of this kind. One of the earliest *car following models* is that of Pipes see [13], [96] and references therein. In this model, the acceleration of a car following a car in front is represented by the following equation:

$$\frac{dv_i(t)}{dt} = \frac{v_i(t) - v_{i+1}(t)}{T}$$

where $v_i(t)$ and $v_{i+1}(t)$ represent the speeds of the following and leading cars, respectively, and T is a relaxation parameter. This differential equation represents a stable system where the following car has a stronger acceleration when the speed difference between the two cars is large. Seeking more realistic representations, this approach was followed by several models including the General Motors nonlinear model or Gazis-Herman-Rothery (GHR) model [97]:

$$\frac{dv_i(t+\tau)}{dt} = \lambda {v_\mathrm{i}}^m(t) \frac{v_i(t) - v_{i+1}(t)}{(x_i(t) - x_{i+1}(t))^l}$$

in this equation, l and m are model parameters, τ was introduced in an earlier model to destabilize car platoons and $x_i(t) - x_{i+1}(t)$, the distance between the cars, was introduced to avoid collisions, see [13] and references therein.

Gipps introduced a model based on the desirable properties of mimicking the real behavior of traffic, with parameters associated with obvious characteristics of drivers (explanatory basis), and with good behavior when the interval of successive recalculations of speed and position is equal to the reaction time [98]. We conclude this section with the model developed by Krauβ [99]. This model is highly efficient computationally and closely represents real life traffic. Under this model, each vehicle will compute a desired speed v_{des} as the minimum from the maximum allowed speed v_{max} , the current speed v increased with the acceleration $a(v)$ required to approach the preceding vehicle or a safe speed v_{safe} computed according to surrounding condition:

$$v_{des} = \min(v_{max}, v + a(v)\Delta t, v_{safe})$$

On each computing cycle, every vehicle will select a random speed around the desired speed, update position and calculate a new safe speed according to the following assignments:

$$v \leftarrow max(0, rand(v_{des} - a\Delta t, v_{des}))$$

$$x \leftarrow x + v\Delta t$$

$$v_{safe} \leftarrow v_p + b(\hat{v})\,\frac{g - v_p\Delta t}{\hat{v} + b(\hat{v})\Delta t}$$

where v_p is the speed of the predecessor car, $b(v)$ is the stopping acceleration and $\hat{v} = \frac{v+v_p}{2}$. We choose SUMO [100] as our microscopic traffic simulator; SUMO uses Krauβ even though it allows for user implementations of other *Car following Models.*

2. *Lane Changing Models.* Another important aspect of *Microscopic Simulation Models* is that of lane changing. An excellent review and taxonomy of *Lane-Changing Models* is presented in [101]. In this publication, they indicate that this important maneuver can be considered a moving obstruction. According to [101], for the case of *microscopic simulation models*, the *lane changing models* can be divided into, *rule*, *discrete choice, artificial intelligence,* and *incentive, based models. Rule based models* use fixed, deterministic rules (avoiding obstacles, speed gaining, etc) to trigger rule lane changes. A gap acceptance

model is then used to determine if the existing gap should be accepted. In d*iscrete choice base models*, three steps are followed: 1) checking lane changing necessity, 2) choice of target lane, and 3) gap acceptance. The decisions at each step are usually based on logit or probit models depending on the step. *Artificial intelligence based models* rely on artificial intelligence approaches, such as artificial neural networks or fuzzy logic. In i*ncentive based models,* the decision of changing lines is intended to maximize driver benefits. The lane changing model used by SUMO is an *incentive based models* that considers lane changing when the lane in use cannot lead to the destination or when benefits can be obtained by lane changing.

3. *Other models.* There are other microscopic models like the *optimal velocity models*, in which the vehicle speed is not only a function of v_p but also of the space headway h_s between vehicles. *Psycho-physiological spacing models* based on *perception thresholds* that make cars appear larger when they are closer, which trigger actions to slowdown or overtake. These two systems present the inconvenience of either being unrealistic or difficult to calibrate. Another approach is the *traffic cellular automata,* which uses discrete time and space where roads are divided in slots of certain length where vehicles fit. This approach is efficient for computer simulations, can represent real-life traffic phenomena and can be set to include learning and psychological aspects. More details about these approaches can be found in [13].

3. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a relatively new *metaheuristic* based on the social behavior of ants. These insects are very efficient in finding the best path/food-source combination. Their interesting methods are based on individuals with simple behaviors that interchange information indirectly by means of chemical compounds (pheromones) they use to mark their paths. Ant colony optimization was introduced by M. Dorigo and colleagues [19], [20], [21], [22], [23]. According to [23], this new metaheuristic was greatly inspired by the behavior of real ants [24], [25], [26]. Great effort has been dedicated to ACO applications in computer science and engineering. Several ACO algorithms have been proposed for the Traveling Salesman Problem (TSP) [27], [28]. ACO approaches for the Job Shop Scheduling Problem (JSSP) and the Flexible Job Shop Scheduling Problem (FJSSP) have applications in controlling mechanical engineering machines [29], [30]. ACO algorithms have been used in civil engineering for economic optimization of reinforced concrete [31]. Research in Ant Colony Optimization has also been conducted in digital image processing [32], electrical engineering where a system has been proposed for environmental/economic dispatch [33], and as well as several routing protocols in computer engineering [34], [35]. More information on ACO applications can be found in [36], [37].

When natural ants are faced with a food source that can be reached only by crossing one of two bridges of different length, as illustrated in Fig. 3, they usually choose the shortest bridge. At the beginning some ants may use the shortest bridge and others may choose the longest one. As the ants walk they drop pheromones to mark the path. Because the ants that started the shortest way will arrive earlier at the food source and will return following the scent on the trail they used while they drop more pheromone on it, the scent on the shortest trail will increase faster. For the ants starting the process, the stronger scent will orient them towards the shortest path. The process is a little bit more complex and is affected by factors such as evaporation, drop frequency modulated by the quality of the food source, system feedback, and sensor imperfection, among others. Evaporation is an important component that helps minimizing the impact of wrong decisions and may also decrease interest on a route in case of food depletion.

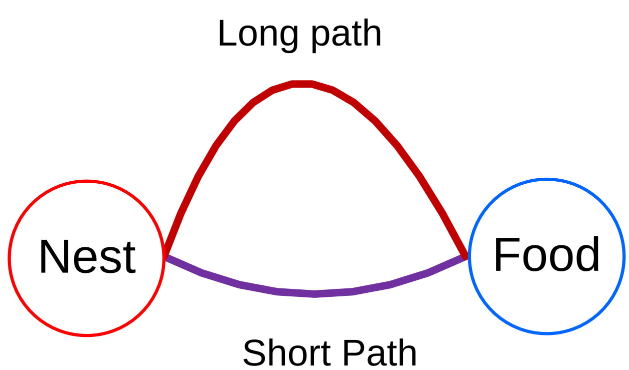


Fig. 3. Path selection of ants.

The path selection mechanism has inspired software approaches with artificial ants for problems where a minimum cost path solution is required both in communications and in road networks [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49]. The artificial ants travel the network and update the nodes information with artificial pheromone that can be sensed by other ants to make routing decisions. It is common that artificial ants mark the information of the arcs they visited only on the way back to the source according to equation (1). In this equation, the pheromone concentration from node i to node j at node i, τ_{ij}, is updated by the traveling ant k with pheromone deposit $\Delta\tau^k$ after evaporation factor ρ is applied.

$$\tau_{ij} \longleftarrow \rho\tau_{ij} + \Delta\tau^k \tag{12}$$

4. AGGREGATION

We can divide the process of road traffic congestion alleviation into an on-demand Traffic Assignment Problem (TAP) and the Data Aggregation mechanism that will trigger the DTA. In this section, we review some useful concepts for any aggregation algorithm.

Enabling vehicles to have real-time access to traffic data is a challenging task due to some of the problem characteristics, such as high mobility, dynamic topologies, potentially unbounded network sizes, and bandwidth constraints. The potentially unbounded network size and bandwidth limitation call for schemes where some form of data compression is used and for efficient algorithms that work on a self-organized, self-monitored and decentralized fashion. Under this circumstances, data aggregation becomes very handy. Data aggregation can be considered a subset of information fusion, that aims at reducing, or summarizing, the handled data volume [102]. VANETs are a fairly new field of research but fortunately they share the need for data aggregation with the more established field of Wireless Sensor Networks (WSN) in which important research on the area has been done. Aggregation is also needed in WSNs, although for the different reason of energy preservation.

Data aggregation is common to database management where some typical aggregate functions are: AVERAGE: Returns the average value from a set of numbers, COUNT: Returns the number of element in a set, DISTINCT COUNT: returns the number of unique values in a set, FIRST: Returns the first element in a list, LAST: Returns the last element in a list, MAX: Returns the largest value in a set of numbers, MIN: Returns the smallest value in a set of numbers, RANGE: Returns the difference between the maximum and the minimum in a set of numbers, SUM: Returns the sum of element in a set of numbers, MODE: returns the value that appears most often in the set of data, among others.

We now proceed to present the definition of aggregation functions and some special types of these functions as they were formally defined in [102].

To ease the understanding of the definition we first make a brief review of important of multiset concepts.

1. A multiset or bag is a generalization of the concept of set in which elements are allowed to appear more than once.

2. The multiplicity n of an element m in a multiset M is the number of times m is contained in M.

3. A sub-multiset is a generalization of the concept of sub-set. In a sub-multiset M of a multiset X, elements in M are allowed to appear more than once but they cannot exceed the multiplicity they had in the original set X.

4. The multiset sum $M = X \uplus Y$ of multisets X and Y, is a multiset composed only of elements of X and Y, that contains all the elements in X and all the elements in Y, and that the multiplicity n of an element m of M satisfies $n = n_X + n_Y$, where n_X and n_Y are the multiplicities of the element m in X and Y, respectively.

5. The support set of a multiset is the set of its different elements.

4.1 Aggregation function

The concept of multiset is handy when defining aggregation as repetitions may be present and order is irrelevant. An aggregation function *f* may be defined as a function that maps a multiset of elements from a domain *I* into an output domain *O*:

$$f: N^I \to O \tag{13}$$

The fact that the input is a multitset implies that the input may have repeated elements and that the order of the input elements is irrelevant.

4.2 Decomposable Functions

The concept of *decomposable* functions is of great importance in distributed systems. When an aggregation function is *decomposable*, it is possible to subdivide the input multiset into several sub-multisets, obtain partial results by applying the aggregation to these sub-multisets and then, combine the partial results to obtain the same result as if the aggregation function was applied to the original multiset. A special case of *decomposable* aggregation functions is that of *self-decomposable* aggregation functions. We now proceed to present definitions for *self-decomposable* aggregation functions and *decomposable* aggregation functions as they were introduced in [102].

4.2.1 Self-decomposable aggregation function. A *self-decomposable* aggregation function f is an aggregation function that for any non-empty multisets X and Y and a merge operator $\diamond$ satisfies the following property [102]:

$$f(X \uplus Y) = f(X) \diamond f(Y) \tag{14}$$

Functions MIN, MAX, SUM and COUNT are *self*- decomposable.

4.2.2 Decomposable aggregation function. A *decomposable* aggregation function *f* is an aggregation function that, for a certain *self-decomposable* function h, and a function g, satisfies:

$$f = g \circ h \tag{15}$$

Examples of this type of functions are AVERAGE and RANGE.

4.2.3 Duplicate-insensitive aggregation function. A *duplicate-insensitive* aggregation function is an aggregation function satisfying $f(M) = f(S)$ for all multisets M with support set *S*. Otherwise, the function is called *duplicate-sensitive.*

There is a huge amount of different approaches on distributed data aggregation algorithms, each with different levels of performance, and taxonomy is of main importance when creating new algorithms. When doing aggregation in distributed systems like VANETs, aggregation is much easier if the aggregation function is *decomposable* and *duplicate-insensitive*. DISTINCT COUNT and MODE are *non-decomposable,* but *duplicate-insensitive* and *duplicate-sensitive*, respectively. *Non-decomposable* functions require the whole set of data available at the moment of computation. Aggregation functions that are both *self-decomposable* and *duplicate-insensitive* allow for the use of idempotent binary operators that could be applied successively by different nodes to the elements of the multiset.

4.3 Taxonomy of Distributed Aggregation Algorithms.

Reference [102] presents a taxonomy of distributed aggregation algorithms based on communication and computation perspectives. The first perspective relates to routing protocols and network topology while the second perspective refers to the aggregation functions computed by the algorithm.

4.3.1 Communication Perspective: According to this taxonomy, the communication perspective is divided according to the routing protocols into *structured*, *unstructured*, and *hybrid* classes. *Structured* algorithms rely on a predefined network topology, such as the categories of *hierarchy* (tree) or *ring*, while *unstructured* algorithms operate without a predefined network topology. Categories in this class include *flooding/broadcast*, *random walk* and *gossip*. Finally, *hybrid* approaches employ a combination of both approaches. Fig. 4 illustrates the Communication Perspective Taxonomy.

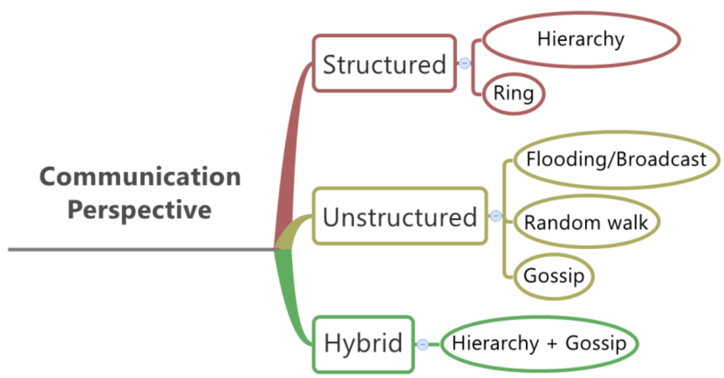


Fig. 4. Communication Perspective Taxonomy

4.3.2 Computation Perspective: This perspective, on the other hand, consists of the following categories: *hierarchical*, *averaging*, *sketches*, *digests*, *deterministic* and *samples*. These categories are intrinsically organized by the underlying aggregation functions: *Decomposable Functions*, *Complex Functions* (allows the computation of any aggregation function), and *Counting* (restricted to the estimation of the COUNT function). *Hierarchical* algorithms allow the computation of any decomposable function. *Averaging* approaches allow for the computation of *duplicate-sensitive* decomposable functions based on the AVERAGE. *Sketches* algorithms allow for the computation of *duplicate-*

sensitive decomposable functions based on the SUM. *Digests* allow the computation of more complex aggregation functions, like quartiles and produce a *digest* that summarizes the system data distribution. *Sampling* algorithms allow the implementation of the COUNT based on probabilistic approaches. Fig. 5Fig. 5 illustrates the Computation Perspective Taxonomy.

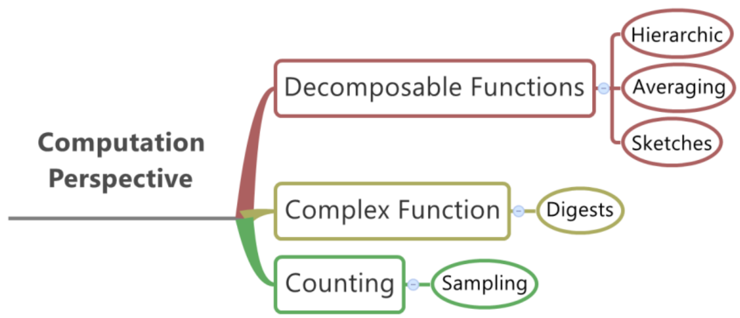


Fig. 5. Computation Perspective Taxonomy

4.4 Some examples of VANET aggregation algorithms.

4.4.1 Cluster-Based Accurate Syntactic Compression of Aggregated Data in VANETs.

In the Cluster-Based Accurate Syntactic Compression of Aggregated Data in VANETs (CASCADE) [103] each vehicle periodically broadcasts its primary record. The primary record consists of timestamp (8 bytes) - the time the record was generated, location (16 bytes) - latitude and longitude, speed (1 byte) - in meters/second, acceleration (1 byte) - in meters/second^2, heading (1 byte) - in degrees from North (0-360), altitude (2 bytes) - in meters above sea level. Primary Records can be re-broadcasted backwards for up to 1.5 km. The primary records representing vehicles ahead of the current vehicle comprise the local view. The local view is divided into 12 clusters with dimensions of 16 m x 126 m. An aggregated cluster record is formed by concatenation of Compact Data Records. A Compact Data Record represents vehicles in a cluster using the differences between the vehicle data and overall cluster data; a compression ratio of at least 86% is achieved in this scheme. Aggregated cluster records are broadcasted periodically and are used by receiving vehicles to form an extended view.

4.4.2 Clustered Gathering Protocol. The Clustered Gathering Protocol (CGP) [104] is a Hierarchical and Geographical Data Collection algorithm with infrastructure support for VANETs. In this algorithm, the road is divided into equal length clusters. Data collected by vehicles is aggregated and sent to service providers by means of Road Side Units (RSU). The process begins with the cluster head election. Each node is the cluster head until it gets a CH_ANNOUNCE message from another node or the cluster head election period ends. Nodes calculate a random back-off time to announce that they are cluster head to their neighbors by using a CH_ANNOUNCE message. This random back-off time is a function of the distance from the car to the end of the segment, which favors cars that are about to enter the segment. During data collection, each node waits a random bounded back-off time. At the end of the back-off time, a node sends a Request to Send (RTS) to the cluster head, the cluster head acknowledges the reception by sending a Clear to Send (CTS), and the node sends its data to the cluster head.

4.4.3 Secure Distributed System inspired by Ant Colonies for Road Traffic Management in Emergency Situations. In this algorithm [105], the vehicles can use the information of previous vehicles to dynamically decide the best path. The algorithm uses generation points at cross road locations. The generation points signal vehicles to send a message with the location ID. These messages are processed similarly to the ants' pheromones in an ant colony. Ant pheromones are substances with scent, which ants use to mark spots they have crossed. The pheromones scent intensity naturally decreases over time and increases every time a new deposition occurs. The pheromones intensity is used by ants to detect highly used paths. In a similar form, the Secure Distributed System inspired by Ant Colonies for Road Traffic Management in Emergency Situations uses it to detect highly used roads chose less congested alternatives at junctions.

4.5 Ant Aggregation

The algorithm that we propose uses two independent aggregation mechanisms, the *pheromone drop* aggregation and the *perceived edge costs* aggregation.

4.5.1 Pheromone Drop Aggregation

Our algorithm is based on the propagation of *pheromone drops,* which are a function of the *aggregated average speed* of the vehicles. This aggregation process is based on clusters that form around accidents. The cluster's members average speeds are used to compute pheromone intensity. The aggregation function that computes the *pheromone drops* is *decomposable* as it is obtained by summing individual contribution (*self-decomposable* aggregation function). The sum is later divided by the COUNT of the reports to obtain the *aggregated average speed*. The *aggregated average speed* is then used in a function to compute the *pheromone drops.* This whole process complies with the definition of a *decomposable* aggregation function. The aggregation process is not *duplicate-insensitive* and for that reason we need to form clusters and avoid secondary *request messages* so that the drop contains no duplicates. This aggregation process emulates the contribution of several ants when they pass over the same spot dropping pheromone.

4.5.2 Perceived Edge Costs Aggregation

Vehicles in a cluster near an accident report the incident in *traffic incident messages.* Vehicles store traffic congestion in a perceived edge cost memory. When a vehicle receives a *traffic incident message*, the *pheromone drop* on that message is stored in the perceived edge cost memory of that vehicle as an indication of the traffic congestion on the associated edge. If a second *traffic incident message* for the same edge is received, the *pheromone drop* of the new message is aggregated (SUM) with the content already in memory. This aggregation process is *self-decomposable* (SUM) but it is not *duplicate-insensitive,* as if two different drops contain duplicated information contributions from the same vehicle on the same road segment, they will affect the aggregation twice. Duplicates are not avoided here, though, a feature that emulates multiple drops of a single ant in the same spot to indicate different quality levels of the food source.

5. LITERATURE SURVEY

Traffic assignment is the process of assigning the *traffic demand* to the routes on a road network. As early as 1850 geographer Johann Georg Kohl analyzed the case of balancing traffic on a two-node, two-route network [53]. Wardrop [4], shows that the distribution of vehicles on a road is simultaneously random in space and time and finds two frequency distributions, one associated with successive vehicles passing a point and the other with successive vehicles along a road at an instant. Wardrop also derives a formula for the frequency with which vehicles would overtake one another when there is no interference with overtaking. He defines capacity and establishes the connection between delay and capacity in relation to traffic signals, also showing that the shortest practicable traffic light cycle does not necessarily result in the minimum average delay. Wardrop's main contributions to traffic assignment are the two equilibrium criteria he introduced. These are included in the problem statement of almost every traffic assignment problem ever since. The two equilibrium criteria for the traffic flowing from a set *origins* to a set of *destinations* are: *user equilibrium assignment* (UE) and *system Optimum Assignment* (SO), depending if we optimize from the user's or system's point of view. These criteria are also referred as *deterministic user equilibrium assignment* (D-UE) and *deterministic system Optimum Assignment* (D-SO). *Traffic assignment* is commonly modeled by means of two different mathematical formulations, the *static traffic assignment* and the *dynamic traffic assignment*, depending if traffic flow is constant or time dependent.

5.1 Seminal works

5.1.1 Static Traffic Assignment works

An STA model of the *equilibrium assignment* for both Wardrop's principles, the UE and the SO, was first presented by Martin Beckmann, Bartlett McGuire and Christopher Winsten (BMW) [5]. This solution became a standard in transportation planning since it was introduced. In the BMW model, the optimality of the UE was guaranteed thanks to the existence of the Karush–Kuhn–Tucker conditions [79], [80], a generalization of Lagrange’s multipliers that introduce some necessary conditions for a solution of an *inequality constrained optimization problem* to be optimal. For more details on STA, BMW, and KKT, please refer to Section 2.2.1. Marguerite Frank and Philip Wolfe presented *an algorithm for quadratic programming* [83]. In [84] the algorithm is adapted for the *traffic assignment problem* and tested in a small city network after which the method was named as the Frank-Wolfe algorithm and became a standard for the next 30 years [85]. The algorithm suffers from slow convergence but is highly efficient in memory usage as it uses aggregation based on the road network links over the *origin destination* pairs, which conveys the highest level of aggregation. Many methods have been proposed using route base aggregation approaches that are less memory efficient. Fewer approaches have been developed on an *origin-based* aggregation approach, of this the Bar-Gera Algorithm has represented a breakthrough of great importance in the field [85]. Currently as new approaches have emerged, solutions based on *dynamic traffic assignment* have become available for dealing with congestion and variable conditions. In [75] the solutions that exceed the capacity of the road and ambiguity of low flow due to low traffic or congested road are addressed in a new model. The SO formulation for this model is very similar to that of the BMW model. The UE presents complications as the existence of the solution is not directly guaranteed as in BMW model, but needs to be determined with the Lagrange multipliers.

5.1.2 Dynamic Traffic Assignment Works

DTA originated from the seminal works of Yagar in 1971 [6], [7], and Merchant and Nemhauser in 1978 [51], [8]. The first dealt with UE using a numeric approach extending traffic estimation methods in [106]. In 1996, a closed expression for the method of Yagar was presented in [107]. The second work was the first attempt to formulate DTA as a mathematical problem, and was limited to the D-SO, *fixed demand*, single destination case [8]. Achieving UE analytically is, in general, a very task unless some simplifications are made. The *Stochastic Network Loading* (S-N-L) type of problems, assumes that the measured travel times are independent from the link flows [9]. According to the authors, S-N-L problems can be approached with analytical methods such as [10] or by stochastic simulation like in [11].

DTA problems approaches can be classified as: *Mathematical programming formulations, Optimal control formulations*, *variational inequalities formulations,* and *Simulation-based models* [8].

5.2 Mathematical Programming Formulations.

Under this category we have the work of Yagar [6], which is the earliest known DTA method [7], and it is an UE approach based on an extension of the method of Homburger [106] by incorporating queues. Homburger described a procedure in which at every time slice, as large a fraction as possible of the whole demand matrix is assigned to the network before the link costs are updated. The link costs are updated, new paths are found, and additional demand is assigned. The procedure continues until all the demand on a traffic slice is assigned. In this procedure, once a link reaches capacity, it is removed from the available links for assignment. Yagar extended this method by not removing the links that reach capacity and by including deterministic queues that would model a bottleneck at the downstream [107]. When updating the link costs, this method takes into consideration the cost of queuing. Even though Yagar presented a numerical method, it can be formulated mathematically [107]. The work of Merchant and Nemhauser also falls under this category. They formulated a discrete-time, non-convex nonlinear programming approach for the SO case of multiple origins and a single destination with a link exit function to propagate traffic [51]. We present more details on this algorithm in Section 2.2.2.1. This problem is later reformulated as a well-behaved convex nonlinear program by manipulation of the exit functions [108]. This publication includes extensions for multiple destinations even though not all of these cases yield convex programs. Multiple destinations require explicit FIFO constraints that, in general networks would not allow convex formulation of the problem. On the other hand, if these FIFO constraints are not used, the solutions would result in unrealistic traffic patterns [8]. In [107] a general and necessary condition to maintain DTA equilibrium based on assigning traffic to the different routes in proportion to their outflows is presented. Interestingly, if this condition is met, FIFO constraints are satisfied. Unfortunately, meeting this condition is not an easy task as it is based on outflows and it does not consider accidents or in link delays. It also requires information on links that may be far away and difficult to reach. Lastly, as we mentioned before, *mathematical programming formulations* also suffer from artificial delays at junctions.

5.3 Optimal Control Formulations

In this type of approach the there is an assumption that the OD trip rates are known; trip rates and link flows are assumed to be continuous functions of time. One example of this type of formulation is in the work of Friesz et.al [109]. In this work, the authors present optimal control formulations for the dynamic traffic assignment of the UE and SO assignments; they are referred to as the Dynamic User Optimization (DUO) and the Dynamic System Optimization (DSO). Remarkably, they present the first generalization of the BMW equivalent optimization for the static assignment as an optimal control formulation. Another example of this type of formulation is presented in [110]. In this publication, the use of elastic demand is considered, which leads to the implicit consideration of departure choices [8]. In [111] the links exit flows functions, which in previous works were defined as functions of the number of vehicles on the links, are replaced with control variables. This allows for generalization of the static UE, which otherwise could be difficult for non-linear exit functions. Optimal control formulations suffer from similar problems as the mathematical program formulations, such as FIFO violations, unrealistic traffic patterns, and artificial delays at junctions.

5.4 Variational Inequalities Formulations

A dynamic generalization of Wardrop's UE is presented in [112]. The authors first analyze the DUE presented in [109] and renamed as Boston Traffic Equilibrium (BTE). The designation as BTE arises from the similarities of this model to traffic in the city of Boston, where frequent changes in link capacities occur (due to road work, weather, etc.). The authors found that when link capacities are more stable and change during certain periods of time, a new generalization is needed. This generalization is known as the

Simultaneous Route-Departure equilibrium (SRD). It requires the costs for the different paths in use, from time of departure to time of arrival, including early/late penalties, to be identical and equal to the minimum path cost than can be achieved from all the route and departure time choices. In [112] for the first time SRD is formulated as a VI approach. VI formulations are, in general, more computational intensive and still suffer from traffic realism issues as the *Mathematical programming formulations,* and the *Optimal control formulations.* It is interesting, though, that this new VI formulation, under certain regularity condition preserves the first in, first out discipline.

5.5 Simulation-based models

5.5.1 Rule Based Approaches.

These type of systems base routing decisions on sets of rules. These rules may be diverse in nature, like *fuzzy logic*, risk or *bounded-rationality* (BR) as described in [113]. In [114] a real time, centralized, information system interacts with users of four different classes in a simulated environment. The classes considered correspond to SO, UE, BR, and PS, where PS users relay on historical or externally specified paths only. In that research, SO and UE users outperform BR and PS users with PS performing as the worst. In general, rule base system do not adapt well in the presence of new situations.

5.5.2 Brute Force Approaches

This type of algorithm assigns traffic by iteratively running complete simulations and updating the link costs at each iteration [8], [115]. An example of this type of algorithm can be found in [116]. In this study they used a $5\,miles\ \times 5\,miles$ of the Dallas-Fort Worth area with traffic demand based on data from the North Central Texas Council of Governments (NCTCOG) between 5 AM and 10 AM. Traffic is assigned to the road network based on initial link costs assigning a fraction of the demand to the lower cost routes recalculating the link costs and iterating until the whole demand has been assigned. The links costs obtained after the simulation, for 15-minute time slots, are fed back into the system, and the process is repeated. They report that the system does not perform well under fast changing traffic conditions, and report oscillations in the traffic flow, a phenomenon that has been studied in detail in [117]. This type of approach is, in general, time consuming and is not based on vehicular communications but in history.

5.5.3 Game Theory Approaches

This category groups algorithms inspired by game theory. Non-Cooperative Game Theory (NCGT) approaches to traffic assignment often result in elegant models from which valuable insight can be obtained [118]. The authors of this work group the NCGT approaches to traffic assignment into four categories depending on the players participating in the game:

1. Games against a demon. These are zero sum games in which the gains of one player equal the losses of the other. There is only one objective function that a driver wants to minimize while the demon wants to maximize it.

2. Games among travelers. These games present the competition among users of a road system where every user that enters reduces the utility for everybody. Wardrop's UE can be modeled as a game in this category.

3. Games between authorities. This type of game can be use to model situations where public transport operators compete to provide services.

4. Games between travelers and authorities. In this type of game the objectives of the travelers are not required to be strictly different from those of the authorities.

As we mentioned before, Wardrop's UE equilibrium has been related to the *Nash Equilibrium* in *Non-Cooperative games* [70], [17]. In other words, any UE solution to the traffic assignment problem could be classified as a game among travelers. In [119], we can find a scenario of games between travelers and authorities where three dynamic Non-Cooperative games between the traffic light system and the users of the road system are presented. The objective is to find a mutually consistent optimal traffic light settings and user equilibrium. They present a Cournot game where travelers and authorities make simultaneous decisions, a Stackelberg game where authorities lead travelers in making decisions and a Monopoly game where authorities alone make decisions to obtain a SO.

5.5.4 Computational Intelligence Approaches

A Fuzzy Logic Based Traffic Junction Signal Controller (FTJSC) for multiple intersections is presented in [120]. This generalized system takes into account the number of consecutive junctions, the number of lanes, the lengths of vehicles, and the lengths of streets. This controller is better than previous systems as it is able to control multiple junctions, integrates each junction status, uses fewer control rules, has a lower inference frequency, and incorporates street lengths. In [121], a centralized *artificial neural* system is introduced. This system takes current information from throughout the traffic system, sensor readings, weather, time-of-day, etc., and produces the timings for all signals in the networks to optimize the traffic flow. The traffic control function is implemented by a neural network for which the internal weights are updated by an on-line training process. Simulation results indicate a 10% reduction in waiting time at intersections. An *evolutionary computation* method is presented in [122]; the method is based on three different types of agents: heterogeneous *travelers*, *centroids*, and *links*. *Centroids* refer to *origin* and *destination* nodes such as residential, neighborhood, or work places. *Travelers* exhibit evolutionary route choices, information is pooled for interchange at *centroids*, and *links* present flow dependent travel times and costs. Based on the new information collected at *centroid*s, travelers make route choices considering their individual value of time and the toll charged by each link segment. This system was compared to other UE and SUE the model proved to be valid and computationally tractable. In Section 5.5.12 we review ACO algorithms, as a special case of Computational Intelligence Approaches.

5.5.5 Approximations to Analytical Algorithms

Algorithms of this type use approximation methods to analytical formulations. Some algorithms of this type are based on the *method of successive averages* applied to the fixed-point problem. Link flow or cost could be average resulting in MSA-FA and MSA-CA algorithms. We can find examples of this type of algorithm in [18] and [123]. In these publications the Method of Successive Averages is used to solve the fixed-point travel assignment problem. In general, these algorithms may suffer from the rigidity of their inspiring analytical model as well from convergence issues [123].

5.5.6 Network State Reaction approaches

Reactive route guidance algorithms are based solely on the current conditions of the network. The system aims at equal travel conditions for the routes of in each OD for the current conditions. This type of algorithm usually exhibits less complexity than its predictive counterpart [91]. Examples of this type of algorithm include [124] where a decentralized reactive feedback system is evaluated. The paper presents a decentralized feedback route guidance strategy for complex, meshed traffic networks using simple control components of the bang-bang, P, or PI types. These components could be the result of a trial-and-error design. The system was simulated for two example networks under several scenarios of demand and incident conditions. Even though exclusively based on measurable instantaneous travel times no predictions, no demand, and no origin–destination information, the system is shown to considerably reduce travel delays compared to the no-control case

5.5.7 Network State Prediction Approaches

This type of system aims at producing equal travel times for the different OD based on anticipated traffic conditions [91]. This type of approach tries to anticipate the network state from history to provide route guidance. Under this category, we can also find DynaMIT [125]. In this publication the authors present a traffic assignment system that uses both historical road network information and real-time information. The system uses one supply and one demand simulator. The demand simulator estimates the OD flows and traveler decisions on departure time, mode, and route choices (the initial OD flows are obtained from historical data). The supply simulator then uses the OD flows to map the OD flows into link flows and a traffic assignment matrix is generated. The traffic assignment matrix and real-time observations are used to refine the OD flows. This process is continued until congruence between the simulators is obtained. As a result of this process, predictive status of the network and prescriptive route recommendations are presented to the drivers. The guidance is deemed consistent if the drivers' decisions do not invalidate the anticipated guidance.

In order to achieve realistic travel decisions, a mix with reactive approaches is often used. One example of this type of mixed approach is presented in [126]. In this publication they report the use of Kalman filters to predict time-dependant traffic demand based on sensors and historical information, to estimate traffic conditions and provide route guidance. In Section 5.5.12.3 ACO Network State Prediction Approaches algorithms [49] and [44] are explained.

5.5.8 Centralized Approaches

In [121], a centralized *artificial neural* system is presented; this system is explained in Section 5.5.4. An ACO based algorithm for DTA that routes vehicles at intersections is presented in [42]; details are provided in section 5.5.12.5. In [46] a centralized system is proposed in which ants going from a given origin to a destination select the 5 best routes and update their pheromone just in those routes. In case of congestion ants choose alternate best routes until the situation is resolved. This algorithm is explained in section 5.5.12.7.

5.5.9 Decentralized Approaches

The previously mentioned *evolutionary computation* method presented in [122] also falls under this category. As mentioned before, the method is based on three different types of agents: heterogeneous *travelers, centroids*, and *links,* where *centroids* refer to *origin* and *destination* nodes. *Travelers* gain information by travelling, information is exchanged at *centroids,* the information obtained is used to make route decisions. This work is explained in 5.5.4. In [48] two strategies to avoid congestion are presented using pheromone marks on a decentralized system of servers; this system is explained in Section 5.5.12.7. [44] presents a DTA approach to anticipatory routing inspired by ACO inspired agents representing vehicles and roads. This algorithm is explained in Section 5.5.12.3.

5.5.10 Approaches with Infrastructure Support

All Centralized algorithms require infrastructure support. As mentioned before, [44] presents a DTA approach based on ACO inspired agents representing vehicles and roads. This system is decentralized but requires infrastructure. In [45] a local control of traffic lights is proposed along with an infrastructure support. This is a cell-organized system, where vehicles use ACO inspired algorithms for routing. [47] presents a DTA system where the road map is divided into zones in a hierarchical way. Vehicles interchange information with infrastructure to allow travel time estimation and best route selection by ACO. More details on these last two algorithms are presented in section 5.5.12.4 and 5.5.12.5, respectively. [49] presents an infrastructure-supported dynamic routing system based on ACO Network State Prediction. Details on this algorithm are presented in Section 5.5.12.3.

5.5.11 Approaches with No Infrastructure Support

Arellano and Mahgoub [63] present a decentralized and infrastructure-less algorithm. This system is the core of our proposed algorithm and it is explained in the next section.

5.5.12 Ant Colony Optimization

5.5.12.1 ACO Seminal Works. *Ant Colony Optimization* was first used in traffic assignment in [38]. The authors used a modified version of *Ant Colony System* (ACS) [27], for traffic assignment. Several ant colonies are used, one for each OD pair, where the origin is the nest and the destination is the food source. In this system ants from a certain colony respond only to the pheromone of their own colony. Link costs are affected by the total flow though. The authors found that the algorithm performs well even in the presence of complex networks. They claim that ACO systems are suitable in almost all real cases where UE is needed without the use of simplifying assumptions. They state that ACO algorithms are particularly suited for parallel processing and dynamic systems because that is precisely the nature of the algorithm. Interestingly, the authors report that in these algorithms the shape of the objective function is irrelevant, and they can be applied successfully to non-separable link cost functions or multi-class demand.

5.5.12.2 ACO Approximations to Analytical Algorithms. An *Ant Colony Optimization* model using multiple ant colonies, one colony for every *origin destination* ($\boldsymbol{od}$) pair is used in [39], where *stochastic user equilibrium* (S-UE) algorithms for the *fixed-point traffic assignment* problem are proposed. We will introduce this problem after introducing some definitions. For a network of n links let

$\boldsymbol{f}$ be the $(n \times 1)$ vector of link flows;

$\boldsymbol{c}$ be the $(n \times 1)$ vector of costs;

$\boldsymbol{N}$ be the number of routes in the road network;

$\boldsymbol{F}$ be the set of feasible link flows;

the *fixed-point traffic assignment* can be formulated as:

$$\boldsymbol{f}^* = \boldsymbol{f}\big(\boldsymbol{c}(\boldsymbol{f}^*)\big) \;\; \boldsymbol{f}^* \in \boldsymbol{F}$$

where $\boldsymbol{f}^*$ corresponds to the optimum traffic assignment.

The authors prove that with proper selection of the intensity of the pheromone drops, the application of the algorithm is equivalent to the application of an MSA algorithm, such as Dial's algorithm [10]. They designate their algorithm as MSA-ANT. Another example of algorithms of this type, *Ant Colony System for Traffic Assignment* (ACS-TA), for both S-UE and D-UE is presented [52]. The authors generalize the algorithm presented in [39] by analyzing several route choice models and how these relate to the shape of the function that determines the concentration of pheromone deposition over time. They indicate that, by changing this function, different models could be obtained, from the classical Logit to the sophisticated Probit. Additional information on transportation fixed-point problems can be found in [12], [18]. Unfortunately, all these algorithms are intended for the STA.

5.5.12.3 ACO Network State Prediction Approaches. *Dynamic routing system based on Ant Based Control* [49], presents an infrastructure supported DTA system that uses ants to compute and predict travel times. This system is based on *Ant Based Control* (ABC) [127]. ABC is inspired by packet-routing algorithm AntNet [128]. The intersections maintain local time tables that list the travel times to the current node from the adjacent nodes and probabilities tables that store probability-based, goodness factors for each next-link destination pair. When vehicles approach a node they are directed to the link with the greatest goodness factor that leads to the destination. The tables are maintained as follows: for each pre-defined period of time they have a historical average of link speeds that is retrieved from the system memory. This information is combined with the current estimation of speed determined by the algorithm according to formula:

$$v_{AB} = \tau h_{AB}(I_k) + (1 - \tau) f_{AB}(I_k).$$

where v_{AB} is the average speed between nodes A and B, $h_{AB}(I_k)$ is the historical speed value at interval I_k, $f_{AB}(I_k)$ is the estimated speed value on the same interval and τ is a combination factor that the authors set to 0.5. The routing tables are updated as explained next. Forward ants are sent periodically from each car to its destination. When arriving at a node, the estimated travel time is determined when the algorithm takes into account the number of forward ants that were present on that link, using a function that relates vehicle density to speed. Forward ants store these estimated travel times in their memory, and once they reach the destination, they transfer their memory to a backward ant and then die. Backward ants return to the originating node, using the same route as the forward ant, and they update the time and probabilities tables on the way back. The simulation of this system on a part of the Dutch highway network showed that, in general, for 50% of the routes, faster alternatives were found. The proposed algorithm suffers from being centralized.

Another example of this type of approach, *Anticipatory Vehicle Routing Using Delegate Multiagent Systems*, is presented in [44] where they develop a DTA decentralized system to provide anticipatory vehicle routing using a multi-agent system. This system is inspired by ACO and includes agents representing vehicles and roads. In a simulated environment of vehicles, each vehicle sends delegated exploration ants to evaluate alternate routes to the destination. Exploration ants estimate travel time by querying the agents on the different road segments composing the routes about the estimated travel time of the segment. When these ants reach the destination, they send the aggregated travel time back to the vehicle. The vehicle then makes a route decision, and sends intention ants along the selected road to update, by means of notifications, the future demand information of the segments. Road segments use these notifications along with real time traffic information to forecast travel time. The authors report improvements on average the travel time of up to 35%. This system has the advantage of being decentralized but unfortunately requires infrastructure.

5.5.12.4 ACO Optimization of Traffic Signal Settings. Frequently in the literature, the Network Design Problem (NDP) for a road network aims at minimizing the total system costs under limited expenditure, while accounting for the route choice behavior of network users by expanding the capacities of the existing congested links or building new links [129]. The Network Design Problem is more general and may include traffic light settings. When it deals with optimizing signal settings is indicated in the literature as the Signal Setting Design Problem (SSDP). When this is done for the entire network, it is referred as the Global Optimization of Signal Settings (GOSS). On the other hand, if it is assumed that signal settings at each intersection depend only on entering flows of the intersection the problem is referred to as the Local Optimization of Signal Settings (LOSS). Both problems can be formulated as a *fixed-point traffic assignment* and LOSS is also referred to as the asymmetric traffic assignment problem.

An ACO approach for solving LOSS is presented in [41]. The authors solved this problem by controlling the traffic lights by means of pressure variables that are assigned to each road direction. These pressure variables are directly proportional to the traffic flow in the associated direction and inversely proportional to the road width. The pressure variables are used to define a control policy, which is solved as an MSA problem. When simulated on real-scale networks the proposed algorithm obtains the solution in less time but with the same accuracy than a traditional MSA.

In a more general context, NDP problems may include simultaneous optimization of signal setting and traffic assignment. This type of problem is characterized by the so-called bi-level structure. Problems of this type, in general, are difficult to solve, because they require the evaluation of an upper-level objective which, in turn, involves solving the lower-level problem for every feasible set of the upper level decisions [130].

Baskan et al. [131], introduce ACO Reduced Search Space (ACORSES), an ACO algorithm that searches for optimal solutions around the best outcome of the previous iteration. It is used to optimize traffic signal timings under the Mutually Consistent (MC) solution for optimizing signal timings or the bi-level solution for optimizing signal timings. In the first case, the traffic light problem is solved by keeping the flows fixed, and then the traffic assignment problem is solved by keeping the signal timings fixed. The second case requires iteratively solving the traffic light problem which, in turn, involves solving the traffic assignment problem for every feasible set of the traffic light decisions. ACORSES was found successful in terms of the signal timings and the final values of degree of saturation with the MC solution requiring more cycle times and being dependent on the initial settings. Unfortunately, this algorithm is intended for STA problems.

Dynamic Vehicular Traffic Control Using Ant Colony and Traffic Light Optimization, a system that includes *VANET based Traffic Light Optimization* and *Vehicular Routing* *with Optimal Path* is proposed in [45]. The road network system is divided in geographical cells. Additionally, the system is organized in three layers: 1) the physical layer, that includes all nodes and links, 2) the junction layer, where all links that connect to a single junction are eliminated, and 3) the inter-cell layer that only shows links that connect two different cells. The nodes on the last layer are designated border-nodes. The system includes a LOSS mechanism and is an infrastructure supported, junction routing mechanism at the border-nodes. The LOSS portion of this system counts the cars approaching a given intersection in each direction. It does so by direct radio contact and by asking the furthest away cars to extend the area and count the cars behind them. The number of vehicles approaching in each direction is then used to control the traffic light at the intersection. Border-nodes keep inter-cell routing tables tracking the numbers of vehicles going to the adjacent nodes on the neighbor cells during specified time intervals. These tables are disseminated over the entire road network. An ACO algorithm is used to find the shortest routes inside each cell. When doing inter-cell travel, the ACO algorithm includes the inter-cell routing tables in the best route selection. The proposed system was evaluated in a simulation and it was determined that average speed increased around 20 km/h and the number of vehicles stopped at intersections decreased significantly. Unfortunately, this system requires infrastructure support.

5.5.12.5 ACO Routing at Intersections. [42] presents an ACO based algorithm for DTA by routing vehicles at intersections. This work identifies the main differences between a vehicle network and an ant network and introduces Ant Colony Routing (ARC), a modified ACO algorithm that addresses the following existing differences between vehicles and ant networks:

- Ants have no individually pre-assigned destinations, while each vehicle in a traffic network does.
- Ants on an ant network only strive for the user equilibrium, while vehicular traffic management has global objectives.
- Ant networks have no limiting capacities on links, while traffic networks are constrained by link capacity.
- In an ant networks link costs are fixed and static, while in road networks they change dynamically depending on time-varying traffic conditions.

The authors address the problem of pre-assigned destination by using multiple ant colonies, one for each OD pair. They propose a centralized system with global objectives, and address the problems of links' capacities and fixed costs by using a system with two types of pheromones, the traditional ACO pheromone, and a stench pheromone that repels ants from congested links. By doing this, the authors take care of link capacities and costs. The recommended splitting rates for each destination are disseminated at intersections. After a simulation on the Walcheren area in the Netherlands, the authors found that the algorithm is suitable for on-line optimization, and balances well control/performance and computational load. Unfortunately, this algorithm is centralized. Centralized algorithms for VANETs require the existence of an infrastructure that may not always be available and suffer from great computational complexity.

Hierarchical routing in traffic using swarm-intelligence is presented in [47]. It is a DTA system where the road map is divided into city zones interconnected by highways. The nodes of a zone that can directly link to a node in different zones are designated routing-nodes. Vehicles interchange travel information with infrastructure to update travel routing tables at each intersection. These tables include a timetable that lists the travel times to the current node from the adjacent nodes and a probabilities table that lists probability-based, goodness factors. This last table stores goodness factor for each next-link destination pair on the current zone, and for the routing-nodes when the destination is in a different zone. Nodes located in other zones are represented in this table as a virtual node that summarizes the neighbor zone. Routing-nodes periodically send exploring ants to neighbor zones in order to update travel time information on the corresponding virtual node and the goodness of this connection on the tables of the routing-node. The best route selection is implemented by means of an adaptation of H-ABC [132], a scalable ant colony optimization algorithm for dynamic routing in packet switch networks, inspired by routing protocol AntNet [128]. In the proposed algorithm, when vehicles reach a node they are directed to the link of greatest goodness that leads to the destination. Simultaneously, forward ants, are sent to find the best route to the destination. Once these ants reach the destination, they die and backward ants are sent, using the same route as the forward ant, to update the time and probabilities tables. After the ants have updated the road network, the route with the lower cost is selected. This system suffers from being centralized and requiring support infrastructure.

5.5.12.6 ACO Vehicle Routing Problems. The *Vehicle Routing Problem* (VRP) can be described as simultaneously determining the routes for several vehicles from a central supply depot to a number of customers and returning to the depot without exceeding the capacity constraints of each vehicle. This problem is closely related to the TSP, which consists of given a list of cities and the distances between each pair of cities, finding the shortest possible route that visits each city exactly once and returns to the origin city.

An *ACO Heuristic for Vehicle Routing Problem* is presented in [43]. In a TSP problem, ants are provided with the list of cities. Sequentially each ant constructs a tour in which the next city to visit is selected as in a regular ACO algorithm with pheromone deposition and evaporation. After a given number of ants m a feasible solution is found by selecting the best route and an additional amount of pheromone, that depends on the tour quality, is added to the arcs of this solution to promote the use of shortest paths. The process is iterated a set number of times and the solution to the problem is the feasible tour with the smallest cost. In [43] the method is adapted to the VRP in an environment where a given number of ants, representing vehicles with limited capacity, departing from a depot visit customers exactly once, to collect packages. The algorithm is modified by having the ants return to the depot when the vehicle capacity is reached or when all customers have been visited. In this way, each single ant finds a solution to the problem. The solution with the smaller total cost is selected. The authors also study the use of multiple ant colonies, one per vehicle, and the use of a candidate list that limits the options of the next customer to visit. When compared to known optimal solutions the algorithm is successful in finding solutions within 1% difference, and in the case of large problems, the use of multiple ant colonies provided a competitive solution technique. The algorithms proposed for this interesting problem, although aimed at making route decisions, are not intended for optimizing traffic.

5.5.12.7 ACO Congestion Avoidance. In [46] a *Dynamic System for Avoiding Traffic Jams* (DSATJ) is proposed. In this system, ants going from a given origin to a destination select the 5 best routes based on traveled distance, using ACO. After the routes are selected and simulation starts, pheromone intensity on each link is updated based on the number of vehicles using the link. When the pheromone intensity reaches a threshold, congestion is detected and, vehicles entering the simulation would change their route if the affected link was included in their tour, while vehicles already on the simulation would divert to another link before they reach the affected one. A normalization formula is used to reduce pheromone intensity as traffic jams decrease. This system is an adaptation of [133], which is an algorithm for the *Dynamic Traveler Salesman problem* (DTSP) based on Ant System [23]. In [133] new routes are required, not because of congestion, but because the link lengths are artificially changed. The proposed system is centralized and requires infrastructure support.

Self-Organizing Congestion Evasion Strategies Using Ant-Based Pheromones is presented in [48]. Two strategies to avoid congestion are presented using pheromone marks on a decentralized system of servers. In this system inspired by ants, a system of local servers keep the transit time, for every link, of each vehicle on the link. This information is used to categorize links as congested or uncongested. Two strategies are presented to handle congestion, *Stay on Track Strategy* (STS), and *Immediate Evasion Strategy* (IES). In the first strategy, vehicles stay on route until the delay exceeds that of an alternate route. In the second strategy, vehicles change routes as soon as congestion is detected. STS works best for short delays and IES does for long delays. Simulation results did not show a clearly winning strategy for mixed environments. In this type of environment, around 20% of the vehicles were able to significantly improve their travel time under congestion. The problem with this approach is that it requires infrastructure, even though it is not centralized.

In Table 1 we illustrate the advantage the algorithm we propose. It is a DTA, decentralized, and infrastructure-less algorithm.

Table 1. Comparison of different ACO inspired algorithms with the proposed algorithm

Algorithm	**DTA**	**Decentralized**	**Infrastructure-less**
ACS [38]			
MSA-ANT [39] and ACS-TA [52]			
Dynamic routing system based on Ant Based Control [49]	X	X	
Anticipatory Vehicle Routing Using Delegate Multiagent Systems [44]	X	X	
ACO-based algorithm for solving the LOSS problem [41]			
ACORSES [131]			
Dynamic Vehicular Traffic Control Using Ant Colony and Traffic Light Optimization [45]	X	X	
ACR, [42]	X		
Hierarchical routing in traffic using swarm-intelligence [47]	X	X	
DSATJ [46]	X		
Self-Organizing Congestion Evasion Strategies Using Ant-Based Pheromones [48]	X	X	
Proposed algorithm, Road-ACO	X	X	X

6. THE PROPOSED ACO INSPIRED DTA ALGORITHM

We propose an ants-inspired DTA algorithm. Ants are known to find shortest path solutions and ACO algorithms have been used before for traffic assignment successfully.

6.1 Proposed Algorithm

We propose a novel algorithm, Reverse Online Algorithm for the Dynamic-Traffic-Assignment Ant-Colony-Optimization-inspired (Road-ACO), that assigns traffic as it evolves in real time, without prior knowledge of the traffic demand or the schedule of the cars that will enter the road network in the future. This novel, decentralized, online algorithm employs a new breed of ants which are position aware, capable of broadcasting pheromone information, have the road map in memory along with perceived edge costs, and execute shortest path algorithms in a selfish manner consistent with S-UE, just like a VANET-enabled car. Additionally, these ants differ from the traditional ants by the reverse way they use pheromone. Higher intensity indicates road segments of lesser quality, in contrast to better routes in traditional ants. Similar to traditional ants, the new breed of ants use pheromone subject to evaporation, and make routing decisions based on the pheromone concentration on the edges, although they make decisions based on the pheromone concentrations on the entire map and not just on the current node. In real life, the ants' pheromone evaporation indicates routes becoming less appealing due to food quality depletion. In our case, evaporation is the mechanism that progressively increases the quality of routes as decreased pheromone concentration means less congestion. Using the terms ant and vehicles indistinctively, starting with the variables definitions in Section 6.1, we now proceed to describe Road-ACO.

6.2 Algorithm Variables

We now define the algorithm variables. The variables can be divided into global and local variables. The global variables are constant and have the same value for all vehicles. Local variables store data corresponding to the individual vehicles.

6.2.1 Global variables

- *Aggregation period (ap).* Duration window that defines the time used by vehicles to determine travel conditions.
- *Edge default travel time (edtt).* The time it takes to travel that edge at the maximum allowed speed.
- ρ. Represents the pheromone evaporation factor.
- *Speed aggregation threshold (sat).* Threshold used to trigger a vehicle to become a cluster head.
- *Consensus threshold (ct).* Threshold that triggers when a cluster head reports a traffic incident.

6.2.2 Local Variables

- *Step counter.* Used as a timer variable, to cycle through the *Aggregation period.*
- τ_{ij}. Represents the pheromone concentration or edge cost from node i to node j at node i, as perceived by the individual vehicle.

- $\Delta\tau^k$. Represents the pheromone intensity used by a cluster head k to mark the edge it is currently in, in case of a traffic incident.

- *Speed moving average (sma).* Stores the modified moving average of the vehicle speed.

- *Aggregated average speed (aas)*. Used by *cluster heads* to aggregate the average speed of reporting vehicles.

- *Aggregated travel time (att)*. Used by *cluster heads* to compute the pheromone intensity; $att = edge\ length/aas$.

6.3 Dynamic Traffic Assignment Algorithm

The DTA algorithm can be subdivided into three sub algorithms: the *speed aggregation*, *cluster*, and the *communication* algorithms. The *speed aggregation* and the *communication* algorithms run concurrently at all times while the *cluster* algorithm is executed only when an abnormal traffic flow condition occurs. Each vehicle in the simulated road network independently executes these algorithms in a decentralized fashion.

6.3.1 The ***speed aggregation*** Algorithm

Fig. 6 illustrates the *speed aggregation algorithm.* When a vehicle enters into the simulated road network, the global variables are read, the *step counter* is set to 1, and the *speed moving average* is set to 0. At every time step thereafter, the vehicle performs the following actions:

1. Applies evaporation to the perceived edge costs according to $\tau_{ij} = \rho\tau_{ij}$. If, for a certain edge, the new τ_{ij} value is less than the *edge default travel time*, then $\tau_{ij} = default\ travel\ time$ is used.

2. Acquires the *current speed* at which it is traveling and updates the *speed moving average* as:

$$sma \leftarrow (sma * (ap - 1) + current\ speed)/ap .$$

3. The *step counter* is incremented by 1.

At the end of its *aggregation period,* the vehicle checks if the *speed moving average* falls under the *speed aggregation threshold*. If this is true, the vehicle will execute the *cluster* algorithm.

6.3.2 The cluster algorithm

This algorithm is executed when the *speed moving average* falls under the *speed aggregation threshold* in the *speed aggregation* algorithm. Fig. 7 illustrates the cluster algorithm. The vehicle that satisfies the mentioned condition organizes a *cluster* on the edge it is currently in by broadcasting a *request message* and becoming the *cluster head*. Each cluster member will reply with a *reply message* containing the member's average speed. During an entire aggregation cycle, the cluster head aggregates the received average speeds sent by the cluster members into *aggregated average speed*. At the end of this cycle, if the ratio of *number of reply messages in consensus* with the low speed condition (*nrmc*) to the *total number* of *reply messages* (*TNRM)* exceeds the *consensus threshold*, a *pheromone drop* is broadcasted in a *traffic incident message* and the cluster dissolves. The *pheromone drop* intensity is calculated according to:

$$\Delta\tau^k \leftarrow (edtt + att * (nrmc - 1))/TNRM$$

The formula above produces *pheromone drop* intensity increasing with the number of *reply messages in consensus*. If the *total number* of *reply messages* is large but the number of *reply messages in consensus* is small, the drop has low intensity as one would expect and could be negligible. The cluster formation process is explained in the *communication* algorithm.

6.3.3 The *communication* algorithm

Fig. 8 illustrates the *communication* algorithm. Vehicles in the simulated road network continuously monitor the communication channel waiting for two different kinds of messages: *request messages* and *traffic incident messages*. Every vehicle that is on the same edge as the *cluster head* and receives the *request message* will send a *reply message* containing its average speed and resets its *step counter* to 1 to prevent broadcasting any new *request message* until the end of the new aggregation cycle. Every vehicle receiving the *traffic incident message* will update the edge cost according to $\tau_{ij} \longleftarrow \tau_{ij} + \Delta\tau^k$ and execute a shortest path algorithm using its internal map and the perceived costs in memory. The vehicle will reroute if a better route is found.

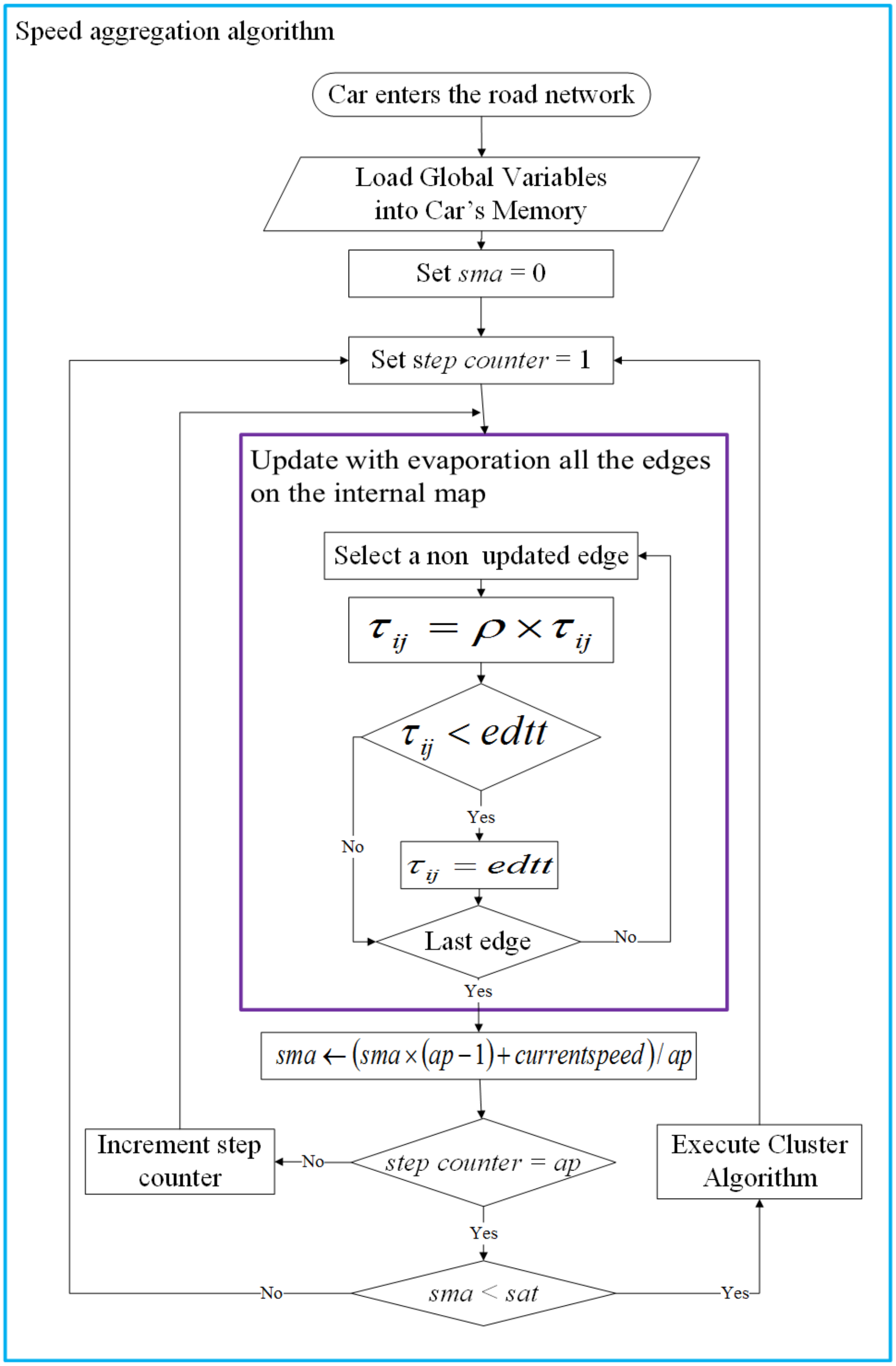


Fig. 6. Speed aggregation algorithm

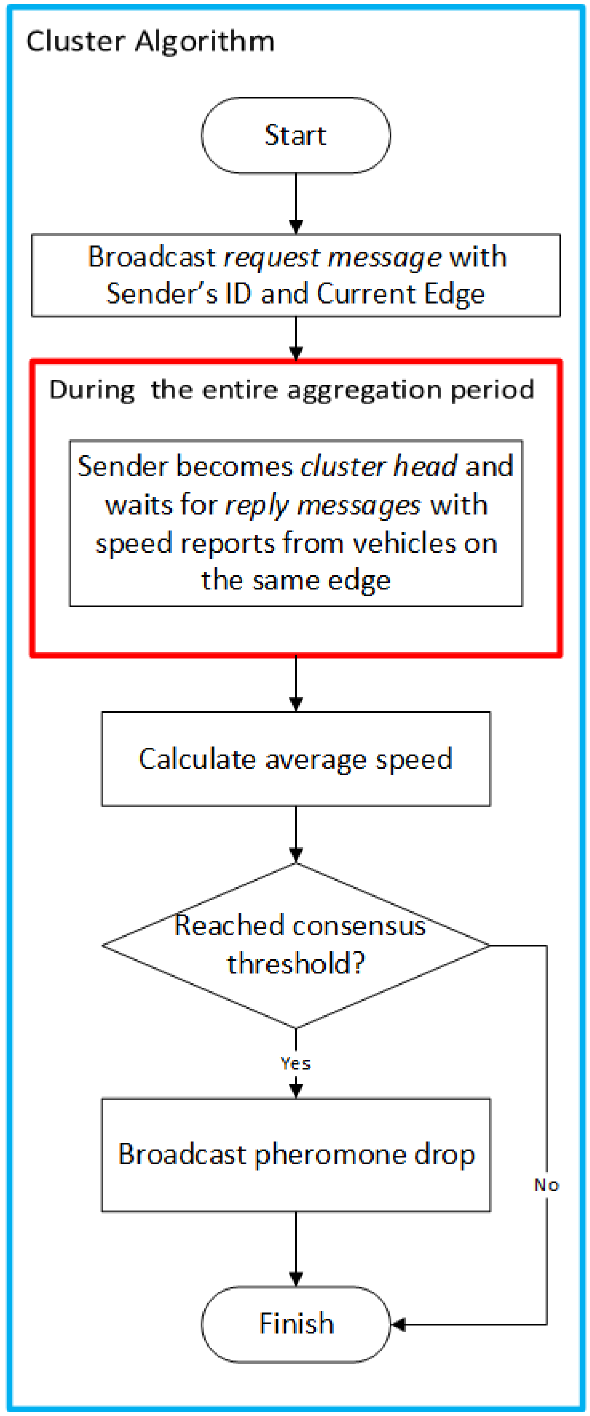


Fig. 7. Cluster Algorithm

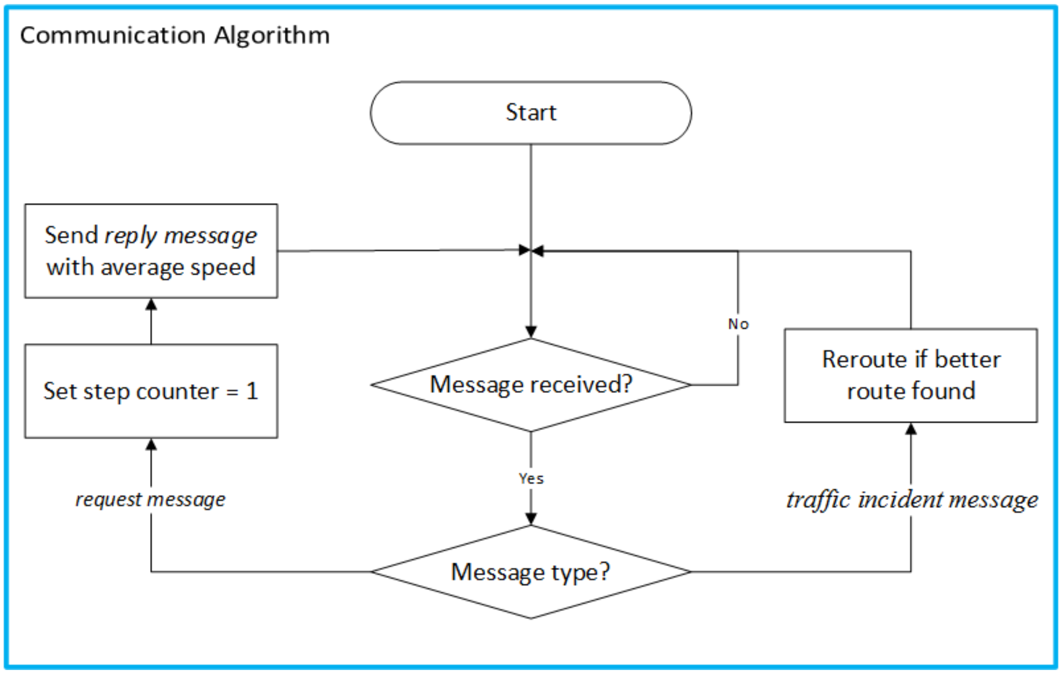


Fig. 8. Communication Algorithm

6.4 SIMULATION RESULTS

6.4.1 TrafficModeler Extensions

We have created extensions to Veins that allow for rapid modeling of traffic demand and VANET simulations [57] using OSM maps. These extensions were successfully tested in a VANET simulation of the Kendal Corridor [134].

6.4.2 Veins Extensions for route changing

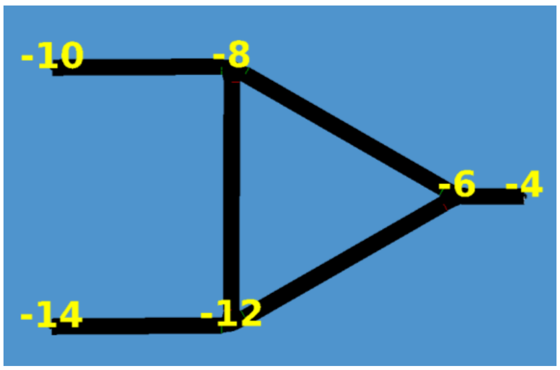
We have extended Veins to allow changing vehicles' routes based on received traffic status messages. The extensions were tested on a rectangular grid with an accident between 2 cars, which, in absence of actions, would create a gridlock from shortly after the accident happened to the end of the simulation. With the extensions and an algorithm that will make vehicles change route if traffic congestion is detected, the gridlock was avoided with an increase of less than 25% on the average travel time (W. Arellano, I.

Mahgoub, and M. Ilyas, "Veins Extensions to Implement Message Based Algorithm for Dynamic Traffic Assignment in VANETs Simulations" [62], in 2014 11th International Conference on High Capacity Optical Networks and Enabling Technologies (HONET) (in press).

6.4.3 Road-ACO

6.4.3.1 Setup. We evaluate the algorithm performance in a simulation environment composed of OMNET++ [135], SUMO [59], and Veins [61], similar to the environment described in [57]. The simulation is performed according to IEEE 1609.4 as implemented in Veins. In the next section, we describe the simulated environment, state the assumptions, and define the input and output parameters.

6.4.3.1.1 **Simulated Environment.** Fig. 9 shows the simulated road network. As we explain below, some vehicles may need to reroute to improve their travel times. All roads have two lanes in each direction, a maximum speed of 64 km/h and U turns are permitted. The horizontal edges are 150 m long, and all other edges are 220 m long. The road network contains 6 nodes: -4, -6, -8, -10, -12, -14. Nodes -10, -14 are origins and node -4 is the only destination. From each origin node, -10, and -14, 1,000 vehicles depart to the common destination -4. When vehicles depart they have a planed route based on a *shortest route* algorithm (SR). Even though this road network is very symmetric and would suggest that all vehicles should use the shortest routes, differences in the traffic light signals and the allowed lane changes make the top route faster and vehicles may need to reroute to improve their travel times. The upper route is more favorable because, first, the traffic lights at nodes -8 and -6 both offer green light in sync to the incoming vehicles, while for the lower route when the traffic light at junction -12 is green, the traffic light at junction -6 has the red light on. Second, the intersection -6 favors the upper edge as the two lanes on it allow travel towards the destination while just one lane from the bottom route allows transit to the destination, since the other lane is forced to turn left. Fig. 10 illustrates these lane details. All the traffic lights in the direction towards the destination have the following cycle: green, yellow, red, equal to 31, 6, and 49 seconds respectively. This was the default cycle assigned by SUMO.



Node Description ($-n$ denotes node n)	
Origins	-10, -14
Destinations	-4
Junctions	-8, -12, -6
Traffic Demand	
OD	Volume
(-10, -4)	1,000
(-14, -4)	2,000

Fig. 9. Road Network Description

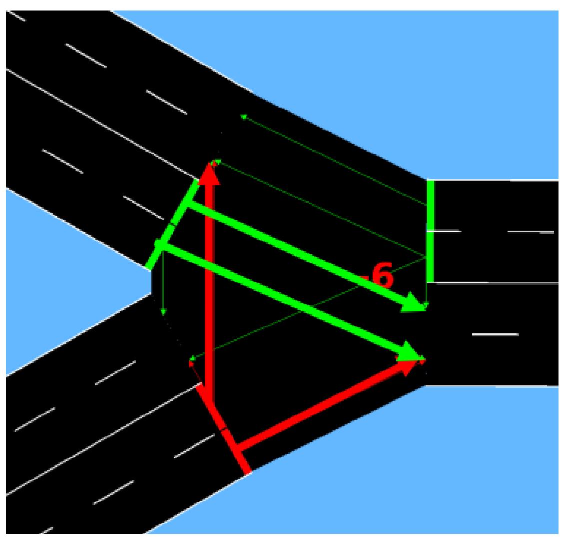


Fig. 10. Lane Connection Detail, intersection -6

6.4.3.1.2 **Simulation Assumptions.** It is assumed that the algorithm messages would be broadcasted using regular IEEE1609 standard beacons. All vehicles are 5 m long and acceleration and deceleration are 2.6 and 4.5 m/s^{2}, respectively.

6.4.3.1.3 **Input Parameters.** The *number of vehicles entering* the simulated road network from each origin. Vehicles enter the network as fast as the congestion of the roads allows. For our simulation we used 1,000 vehicles per origin. This number was selected based on reasonable simulation time and because it produces significant traffic congestion when routing by SR.

The *evaporation factor*. Several values of this parameter are tested to assess the impact on the solution of the speed at which pheromone vanishes when recovering from wrong decisions and congestion. The value of this variable was systematically changed to obtain the greatest traffic flow improvement.

The *speed aggregation threshold.* This parameter is varied to assess the impact of ignoring certain values of congestion level over the solution. The value of this variable was systematically changed to determine how much of congestion can be ignored and still obtain significant traffic flow improvements.

6.4.3.1.4 **Output Parameters.** The outputs of the simulation are the vehicle's *average traveling time,* and the *total time,* measured from the start of the simulation to the moment that the last vehicle reaches its destination. The values of these two parameters are obtained for the cases of route selected by SR, UE as calculated by traffic simulator SUMO (SUMO-DTA), and with the proposed algorithm for several different input parameters. SUMO-DTA is an algorithm that requires entire knowledge of the traffic demand, simulates the scenario, computes new edges' costs and use them in a new simulation, for a given number of iterations. The Veins framework, in charge of controlling the simulation, keeps record of many and diverse simulation results. From there we were able to obtain our parameters of interest, *average traveling time* and the *total time*. In the next section the simulation results are analyzed.

6.4.3.2 Results and Analysis. valuating the performance of a DTA algorithm under UE, by using the definition of UE, is a difficult task as this equilibrium implies that no vehicle can improve its travel time by changing routes and that is hard to assess. We base our evaluation on *average trip times.* A solution that seeks UE should have *average trip times* close to that of the optimal UE. Otherwise, some vehicles could be able to reduce their individual travel times, contradicting the definition of UE. Our algorithm does not minimize *average trip times* as that would produce SO and not UE. However, it is known that UE, even though less efficient, is practical to achieve and not so far from SO [17]. Table 2 contains the simulation results for an aggregation period of $ap = 5$ seconds and a consensus threshold of $ct = 25\%$. The first value is chosen to keep the simulation execution time low, as we observe that lower values increase this time considerably. The second parameter is chosen to allow for early report of the accidents without the need to wait for a large consensus. Row 1 contains the data for the case when shortest route algorithm is used; it is the worst case as it shows the largest *simulation* and *average trip times*. On the other hand, row 2 shows the data for the case of SUMO-DTA. This solution presents the best *simulation* and *average trip times* with an improvement of 34.97% in *average trip times.* SUMO-DTA provides the best results, but unfortunately, requires full knowledge of the traffic demand, a requirement which is not practically feasible. The rest of the rows show the data for the proposed algorithm for different input parameters. All of the proposed algorithm trials show improvement over the *average trip times* of the SR. If any two solutions present similar *average trip time,* we prefer the one with the lowest simulated time. The best outcome is 29.17% for an *evaporation factor* of 75% and *speed aggregation threshold* of 50%. Evaporation factor of 75% appears to be a good choice in this scenario as the three best outcomes include this value. The best solution has a *speed aggregation threshold* value of 50%, which implies less use of the communication channel and prevents unnecessary rerouting.

Table 2. Simulation Results. c*t* =25%, *ap* = 5

Row	**Algorithm**	**Evaporation Factor (%)**	**Speed Aggregation Threshold (%)**	**Simulated Time (seconds)**	**Average Vehicle Trip Time (seconds)**	**Improvement (%)**
1	**SR**	**N/A**	**100**	**7,803**	**278.25**	**0.00**
2	**SUMO-DTA**	**N/A**	**100**	**4,142**	**180.95**	**34.97**
3	**ACO**	**75**	**50**	**5,426**	**197.09**	**29.17**
4	**ACO**	**75**	**100**	**5,229**	**197.31**	**29.09**
5	**ACO**	**75**	**25**	**5,306**	**200.38**	**27.99**
6	**ACO**	**80**	**100**	**5,345**	**203.88**	**26.73**
7	**ACO**	**75**	**75**	**5,476**	**204.51**	**26.50**
8	**ACO**	**50**	**100**	**5,647**	**207.13**	**25.56**
9	**ACO**	**95**	**100**	**5,598**	**213.16**	**23.39**
10	**ACO**	**90**	**100**	**5,663**	**218.89**	**21.33**

7. APPLICATION TO LARGER ROAD NETWORKS

7.1 Applications to Complex Networks

The current version of the algorithm has been tested in a road network that is simple since it only has 5 nodes and 6 branches. Also, this network is small since it has 3 branches of length 220 m and 3 of length 151 m. This is a favorable environment for a VANET system as all vehicles in a branch can be reached by messages from any car on the same branch without the need for rebroadcasting. We intend to test the algorithm in a complex environment with a larger number of nodes and with branch lengths that make it impossible for a message to always reach all the vehicles in the branch without repetition.

7.2 Defining the Complex Network

Fig. 11 illustrates an artificial road network with 4 avenues running vertically and 3 streets running horizontally. Avenues are designated A, B, C, and D from left to right, and streets are designated 1 though 3 from top to bottom. The spacing between streets and avenues of this grid is changed in different experiments according to the following combinations (125, 500), (250, 1,000), (437.50, 1750), (500, 2,000), (1,000, 4,000), and (2000, 8000). The first number in the previously mentioned pairs indicates the distance between streets and the second one is the distance between avenues; all distances are measured in meters. For simplicity we will identify them by the second number in the pair only; for example, we will name the system with streets and avenues distances of 1000, and 4000 meters as the "4000 System". In these road systems the distance between avenues is always 4 times the distance between streets; this is selected so each of them is a scaled version of the (1000, 4000) road system. The appended road segments in the perimeter are the only ones that connect to a single node. These segments have a fixed length of 1,000 m for all systems; this selection is made to allow for flawless insertion of the traffic demand in the yellow zones and keep the demand constant in all cases. It has been observed that the rate at what vehicles are inserted can be affected if these appendices are changed in length. The maximum speed is 100km/h and the roads have 4 lanes, 2 in each direction. Traffic demand is modeled using trafficmodeler [58] with the extensions proposed in [57]. The traffic is generated with random origins inside the yellow left ellipse and travels towards random destinations in the yellow right ellipse. Although the traffic generated this way moves from left to right, individual vehicles may travel in the opposite direction if required by the algorithm. No traffic demand is modeled from right to left. An accident with a duration of 3,300 seconds is simulated near the intersection of street 3 and avenue D as illustrated by the circle and its zoomed image in Fig. 11, where the red vehicles are involved in an accident that produces a total blockage of the road segment in an unfavorable end of a road segment, near the exit node.

With the exception of the 500 System, the traffic demand consisted in all cases of 1800 vehicles entering the road randomly in a period of time of 2 hours. The 500 System uses 900; this is selected because the small grid size would generate a gridlock in the road traffic simulator Sumo. Gridlocks in Sumo can be handled using "vehicle teleportation", this feature would remove vehicles if they are stuck in the same location for a long period of time. We have disabled this feature as it would interfere with the long-lasting accidents that we use in our simulations. Fig. 12 illustrates the mentioned grid lock.

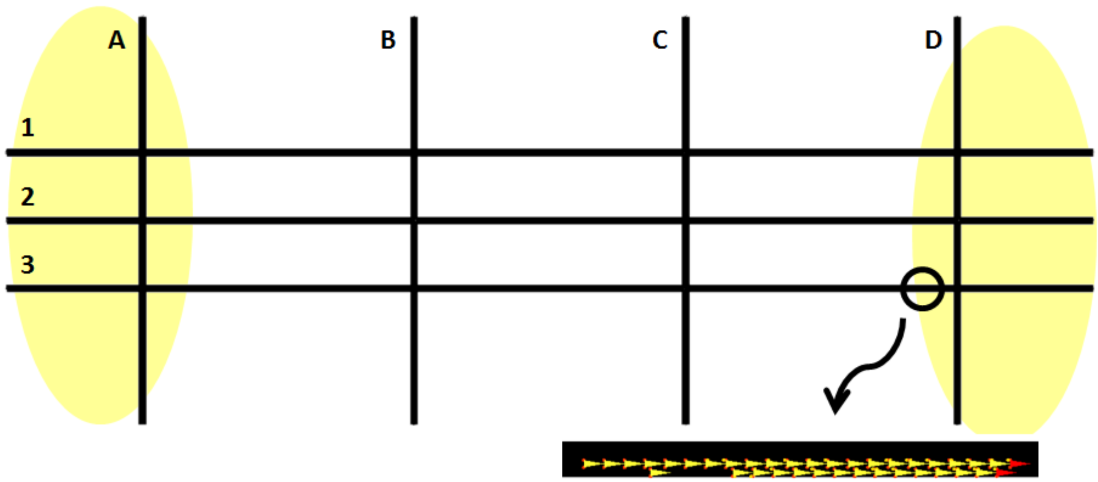


Fig. 11. Complex Road System

Fig. 12. Grid Lock in the 500 System

In the next section we explain how rebroadcasting is used to improve traffic on large road networks where accidents occur far away from the entrance of a road segment.

7.2.1 Modified Algorithm

VANET transceivers are expected to transmit signals with a maximum reach of 1,000 meters. Some of our road systems are larger than that and an accident happening near the exit of a long segment could not reach vehicles entering the road segment and therefore, traffic congestion would possibly not be avoided. To extend the use of the algorithm to large roads we introduce rebroadcasting. Once a traffic accident is reported, vehicles receiving this message will prepare for rebroadcasting by scheduling an individual rebroadcasting time. Any vehicle holding a particular scheduled message of this kind would cancel it, if it detects that another vehicle rebroadcasted it first. The time to rebroadcast (tTR) is a function of the edge length where the receiving vehicle is located (eL), the receiving vehicle position in that lane (rVP) and the accident position (aP). eL and rVP are illustrated in Fig. 13. If the receiving vehicle is in the same road segment of the accident tTR is calculated according to equation (16). From this equation we can see that:

1) the further away from the accident the smaller tTR is,

2) if the receiving vehicle is in the same location as the accident the maximum value of $1/\alpha$ is achieved for tTR. We use $\alpha = 0.10$ in all our simulations.

3) two vehicles at the same distance from the accident, but on opposite sides of it would have identical tTR.

$$tTR = \frac{1}{(|rVP - aP| + \alpha eL)/eL} \tag{16}$$

Simulations results may change if parameter α is varied. If it is too small, tTR could become too large for all vehicles in the edge of the accident in cases when they are all are near the accident. Therefore, the first accident message would take long to be broadcasted, and some vehicles would receive it too late to avoid entering into the segment with the accident.

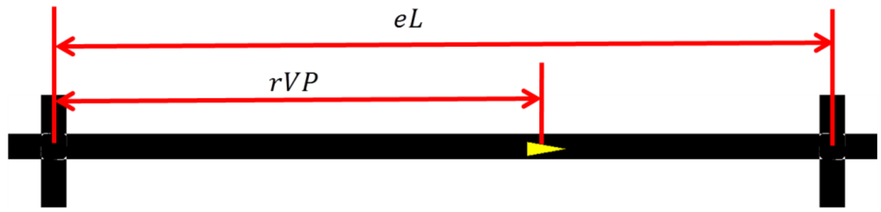


Fig. 13. Distance Meassurements

On the other hand, if the receiving vehicle is on a road segment different from that of the accident, then tTR is calculated using equation (17). This equation will favor those vehicles nearer the intersection, further away from the accident, to retransmit earlier.

$$tTR = \frac{1}{(eL - rVP)/eL} \tag{17}$$

In the following section we present the simulation results. We test the algorithm both with rebroadcasting and without rebroadcasting for two values of the aggregation period. The aggregation period, as defined previously, is the time window used by the vehicles to evaluate traffic conditions before forming a cluster and request for a report. We will show that this parameter is key when redistributing traffic on the roads.

7.3 Simulation Results

In this section we analyze the results of the algorithm with and without rebroadcasting for two values of the aggregation period, 10 and 2 seconds, and 20 values of the pheromone evaporation factor, from .05 to 1.00.

7.3.1 Simulation Results for Aggregation Period of 10 Seconds

In the road systems that we study, the maximum speed is set to 100 km/h or equivalently 27.78 m/s. When the aggregation period is 10 seconds the time it takes to form a cluster and report an accident may be long. For example, in our case the two vehicles involved in the accident are the first two to cross that segment. When one of these vehicles detects the problem and requests a report, it will wait for the 10 seconds of the aggregation period to receive reports of upcoming vehicles. In this time vehicles at a distance of 278 m or less from the entrance node to this segment may enter the troubled road segment. The situation may be worse if we consider that the incident is not reported in the first aggregation period. This may happen because vehicles approaching the accident may still be moving freely and affect adversely the consensus that a problem exists and more aggregation periods may be needed to produce the accident report. As we will see in the following sections, this may negatively affect the effectiveness of the algorithm. In all cases Average Travel Time improvement is calculated with reference to the case of no algorithm.

7.3.1.1 The 500 System with Aggregation Period of 10 Seconds. In Table 3 the results from the simulation for the 500 System are presented, for both, the rebroadcasting and non-rebroadcasting algorithms, with an aggregation period of 10 seconds. In Fig. 14 these results are presented in graphical form.

It can be seen in Fig. 14, that there is no need for rebroadcasting as the non-rebroadcasting algorithm produces comparable results for values of the pheromone evaporation factor in the range of .70 to .85. The algorithms behave differently for values of evaporation factor near 1. This can be interpreted as follows: when rebroadcasting, in the presence of a long lasting accident, if the right information is propagated throughout the network with low or no decay, the knowledge of a long lasting accident can be used efficiently and far away vehicles would reroute efficiently. On the other hand, in the absence of rebroadcasting, as only nearby road segments are influenced by the accident or other perturbations, short term perturbations could impair the algorithm if they are not allowed to decay.

Table 3. 500 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 10 seconds

System	500 No Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	802.19	0.00
0.05	752.84	6.15
0.10	750.75	6.41
0.15	750.10	6.49
0.20	749.07	6.62
0.25	751.61	6.30
0.30	751.61	6.30
0.35	751.79	6.28
0.40	751.79	6.28
0.45	742.31	7.46
0.50	757.74	5.54
0.55	733.06	8.62
0.60	730.72	8.91
0.65	726.40	9.45
0.70	720.59	10.17
0.75	701.81	12.51
0.80	694.85	13.38
0.85	687.89	14.25
0.90	708.19	11.72
0.95	689.96	13.99
1.00	745.85	7.02

System	500 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	802.19	0.00
0.05	715.79	10.77
0.10	709.47	11.56
0.15	745.77	7.03
0.20	751.08	6.37
0.25	736.71	8.16
0.30	732.99	8.63
0.35	735.32	8.34
0.40	732.68	8.66
0.45	733.70	8.54
0.50	723.46	9.81
0.55	722.32	9.96
0.60	724.32	9.71
0.65	716.33	10.70
0.70	745.36	7.08
0.75	738.65	7.92
0.80	735.75	8.28
0.85	732.19	8.73
0.90	714.12	10.98
0.95	683.87	14.75
1.00	682.07	14.97

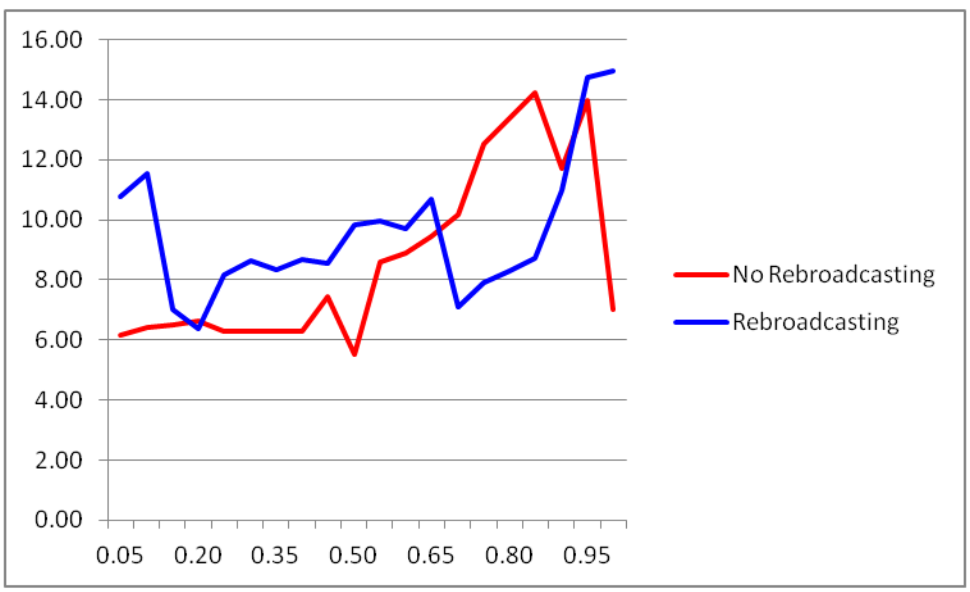


Fig. 14. 500 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 10 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.1.2 The 1000 System with Aggregation Period of 10 Seconds. In Table 4 the results from the simulation for the 1,000 System are presented, for both the rebroadcasting and non-rebroadcasting algorithms, with an aggregation period of 10 seconds. In Fig. 15 these results are presented in graphical form. It can be seen in Fig. 15, that the rebroadcasting algorithm is better than the non-rebroadcasting algorithm for most values of the evaporation factor. This is an expected result as the size of the road system is comparable to the reach of the VANET transceivers. However, there is still no need for rebroadcasting as the non-rebroadcasting algorithm produces comparable results for values of the pheromone evaporation factor in the range of .70 to .85 as in the case of the 500 System. This is a borderline system where still bandwidth savings can be obtained by not using rebroadcasting and selecting the right evaporation factor value.

Table 4. 1000 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 10 seconds

System	1000 No Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	997.04	0.00
0.05	981.42	1.57
0.10	985.33	1.17
0.15	959.41	3.77
0.20	958.84	3.83
0.25	939.66	5.75
0.30	940.03	5.72
0.35	941.63	5.56
0.40	943.58	5.36
0.45	930.83	6.64
0.50	932.89	6.43
0.55	928.85	6.84
0.60	911.18	8.61
0.65	924.91	7.23
0.70	887.31	11.01
0.75	852.13	14.53
0.80	865.51	13.19
0.85	863.53	13.39
0.90	849.50	14.80
0.95	856.73	14.07
1.00	841.38	15.61

System	1000 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	997.04	0.00
0.05	888.33	10.90
0.10	885.13	11.22
0.15	888.10	10.93
0.20	879.74	11.76
0.25	885.67	11.17
0.30	857.07	14.04
0.35	834.17	16.34
0.40	860.00	13.74
0.45	886.31	11.11
0.50	833.31	16.42
0.55	893.40	10.39
0.60	874.71	12.27
0.65	886.31	11.11
0.70	869.25	12.82
0.75	833.31	16.42
0.80	839.14	15.84
0.85	846.70	15.08
0.90	856.80	14.07
0.95	857.19	14.03
1.00	850.18	14.73

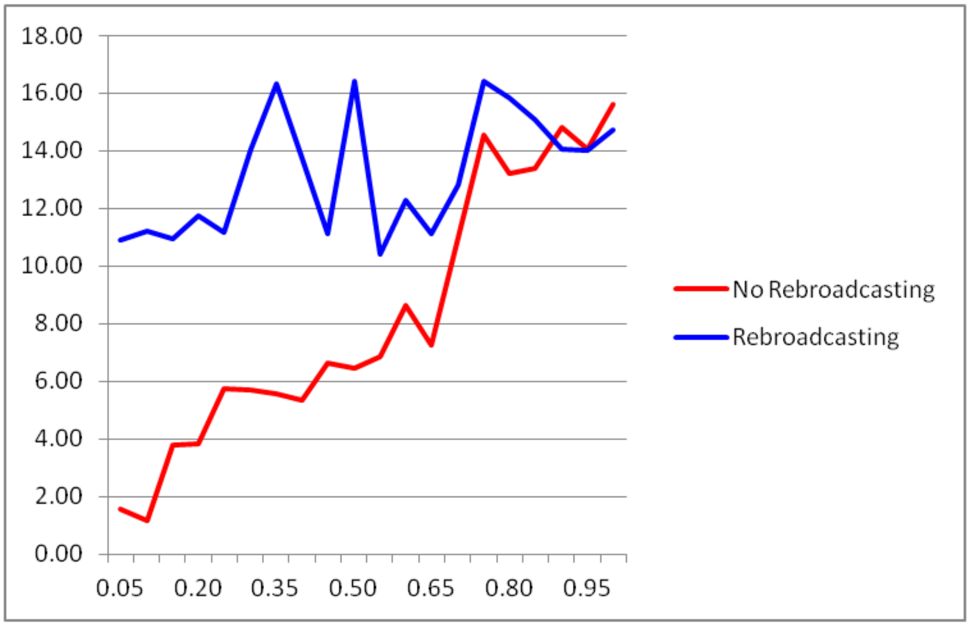


Fig. 15. 1000 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 10 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.1.3 The 2000 System with Aggregation Period of 10 Seconds. We can see in Table 5 the results from the simulation for the 2,000 System for both the rebroadcasting and non-rebroadcasting algorithms, with an aggregation period of 10 seconds. In Fig. 16 these results are presented in graphical form. It can be seen in Fig. 16, that the rebroadcasting algorithm produce results significantly better than the non-rebroadcasting algorithm. For the case of the non-rebroadcasting algorithm, this simulation illustrates the expected result that this scheme would not work for road segment with lengths exceeding 1,000m. This is due to the limited range of the VANET transceivers. However, when using the rebroadcasting algorithm with the aggregation period of 10 seconds in this case, we observe that the efficiency of the algorithm decreases when the evaporation factor reaches 1 and the presence of a peak around the evaporation factor of .70. We present interpretation for these observations in the following section. As the 2000 System situation is unique, we evaluated road segments around it at 1750, 2500, and 3000 meters. The most significant case was the 1750 System which we will explain in the next section.

Table 5. 2000 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 10 seconds

System	2000 No Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	1036.03	0.00
0.05	1039.09	-0.30
0.10	1040.88	-0.47
0.15	1039.09	-0.30
0.20	1038.77	-0.26
0.25	1038.59	-0.25
0.30	1038.67	-0.25
0.35	1038.67	-0.25
0.40	1038.82	-0.27
0.45	1040.88	-0.47
0.50	1039.65	-0.35
0.55	1039.65	-0.35
0.60	1039.65	-0.35
0.65	1039.65	-0.35
0.70	1039.65	-0.35
0.75	1042.61	-0.63
0.80	1039.73	-0.36
0.85	1034.88	0.11
0.90	1031.55	0.43
0.95	1029.54	0.63
1.00	1030.02	0.58

System	2000 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	1036.03	0.00
0.05	1044.73	-0.84
0.10	1034.42	0.16
0.15	1032.30	0.36
0.20	1039.05	-0.29
0.25	1039.05	-0.29
0.30	1037.63	-0.15
0.35	1037.72	-0.16
0.40	1041.75	-0.55
0.45	1037.84	-0.17
0.50	1039.07	-0.29
0.55	1042.45	-0.62
0.60	1037.71	-0.16
0.65	977.55	5.64
0.70	978.67	5.54
0.75	1039.01	-0.29
0.80	1014.80	2.05
0.85	1045.20	-0.89
0.90	1029.59	0.62
0.95	1012.86	2.24
1.00	1038.56	-0.24

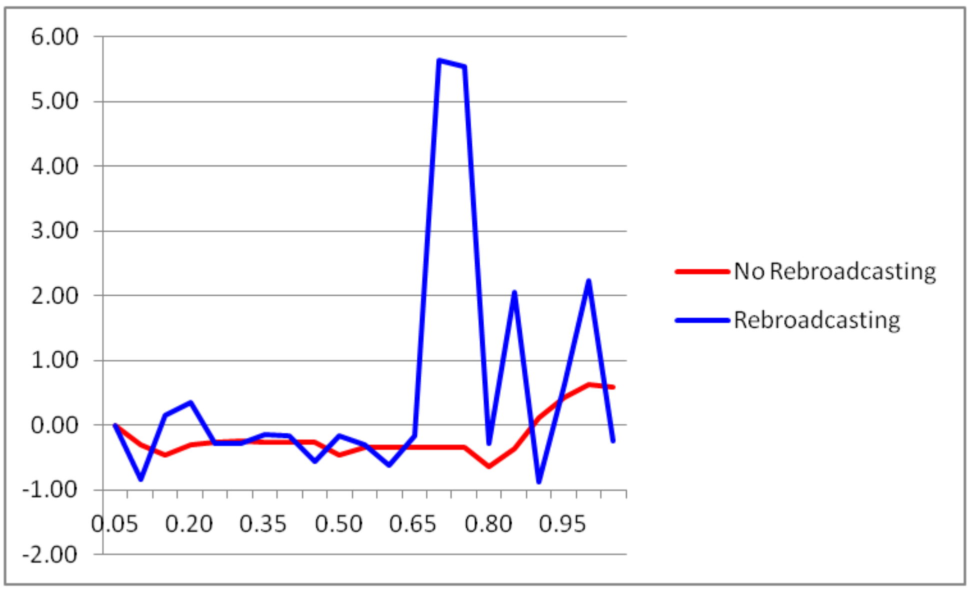


Fig. 16. 2000 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 10 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.1.4 The 1750 System with Aggregation Period of 10 Seconds. In Table 6 we show the results from the simulation for the 1,750 System, for the rebroadcasting algorithm only, with an aggregation period of 10 seconds. In Fig. 17 these results are presented in graphical form. It can be seen in Fig. 17, that there is little gain on the average travel time for this algorithm. This situation can be explained as follows: we have an accident early in time when roads are still being populated. The accident has a long duration, and because of the lengths of the roads, a relatively large number of vehicles must enter the troubled road to allow for the messages about the accident to propagate outside the segment. Because of the size of this road, and the previously mentioned fact that an aggregation period of 10 seconds would not allow many vehicles to receive the information with enough time to react, the number of vehicles trapped in the road is close to the number of vehicles trapped in that road when no algorithm is used.

When we analyzed the 2000 System we noted that the efficiency of the algorithm decreases when the evaporation factor reaches 1 and the presence of a peak around the evaporation factor of .70. In the 1750 System we also observe a diminishment in the efficacy of the algorithm when the evaporation factor approaches 1. We interpret these observations as follows: if the road segment with the accident is already full, and because there is no other long lasting accident, making the evaporation factor large would make transient traffic congestions appear to be permanent, and the route changes could result on making the general traffic condition worst. When the road segment is increased slightly this effect would be still valid. On the other hand, an evaporation factor of 0 disregards the historical information and also results in poor performance of the algorithm. There is no other option for the algorithm to improve other than as a peak as shown in Fig. 16.

Table 6. 1750 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 10 seconds

System	1750 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	911.23	0.00
0.05	906.21	0.55
0.10	905.08	0.67
0.15	905.80	0.60
0.20	905.80	0.60
0.25	905.80	0.60
0.30	905.80	0.60
0.35	905.70	0.61
0.40	906.07	0.57
0.45	906.07	0.57
0.50	906.07	0.57
0.55	906.07	0.57
0.60	906.07	0.57
0.65	906.38	0.53
0.70	905.68	0.61
0.75	907.38	0.42
0.80	900.51	1.18
0.85	907.38	0.42
0.90	912.41	-0.13
0.95	912.38	-0.13
1.00	907.38	0.42

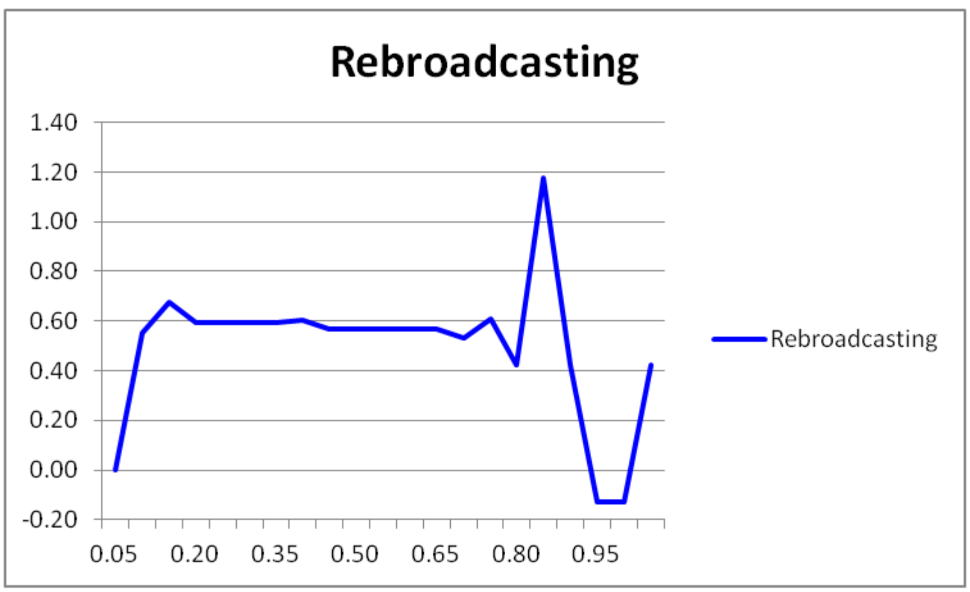


Fig. 17. 1750 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 10 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.1.5 The 4000 System with Aggregation Period of 10 Seconds. We can see in Table 7 the results from the simulation for the 4,000 System, for both the rebroadcasting and non-rebroadcasting algorithms, with an aggregation period of 10 seconds. In Fig. 18 these results are presented in graphical form. It can be seen in Fig. 18, that the rebroadcasting algorithm is consistently better than the non-rebroadcasting algorithm. For the case of the non-rebroadcasting algorithm, it is shown once more that this scheme does not work for road segments with lengths exceeding 1,000m. However, when using the rebroadcasting algorithm with the aggregation period of 10 seconds in this case, we observe that the efficiency of the algorithm increases when the evaporation factor approaches 1. This may be explained as follows: in the presence of a long lasting accident, if the accident information is able to propagate outside the road segment with the accident, making the information about the accident persistent in the vehicles' memory, is useful to improve traffic. This is similar to how humans react in real life. If an accident in a very long road segment with no u-turns is reported, it is convenient to avoid that road and take an alternate route, even if no updates about the accident are received.

Table 7. 4000 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 10 seconds

System	4000 No Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	1269.55	0.00
0.05	1268.97	0.05
0.10	1268.97	0.05
0.15	1268.97	0.05
0.20	1268.97	0.05
0.25	1265.99	0.28
0.30	1268.97	0.05
0.35	1268.97	0.05
0.40	1268.97	0.05
0.45	1268.97	0.05
0.50	1268.97	0.05
0.55	1268.97	0.05
0.60	1268.97	0.05
0.65	1265.99	0.28
0.70	1268.97	0.05
0.75	1268.97	0.05
0.80	1268.97	0.05
0.85	1268.97	0.05
0.90	1261.08	0.67
0.95	1253.15	1.29
1.00	1265.93	0.28

System	4000 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	1269.55	0.00
0.05	1251.66	1.41
0.10	1224.11	3.58
0.15	1250.80	1.48
0.20	1228.98	3.20
0.25	1217.41	4.11
0.30	1268.83	0.06
0.35	1249.96	1.54
0.40	1247.29	1.75
0.45	1241.14	2.24
0.50	1250.48	1.50
0.55	1250.09	1.53
0.60	1235.99	2.64
0.65	1248.16	1.68
0.70	1249.53	1.58
0.75	1239.91	2.33
0.80	1227.06	3.35
0.85	1236.17	2.63
0.90	1227.09	3.34
0.95	1200.75	5.42
1.00	1046.24	17.59

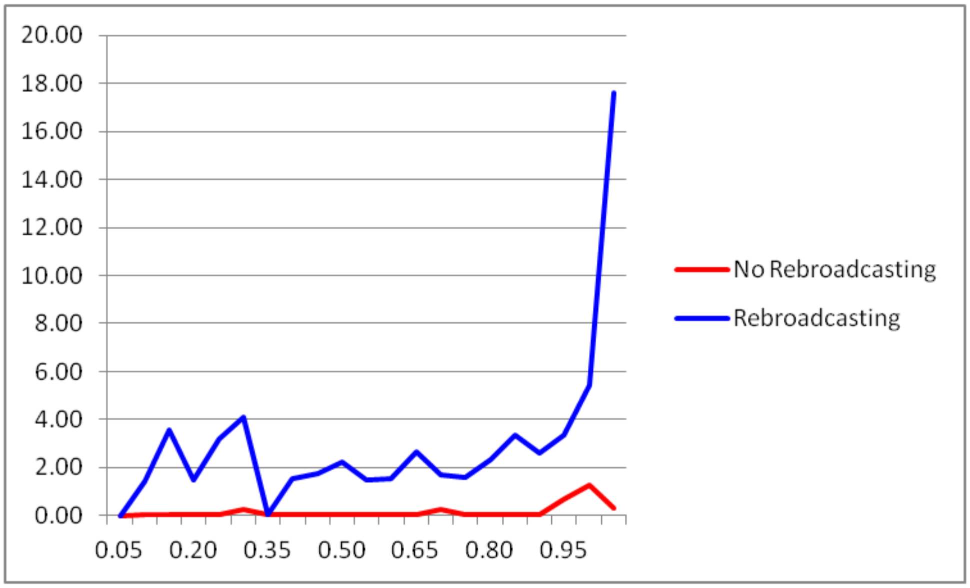


Fig. 18. 4000 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 10 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.2 Simulation Results for Aggregation Period of 2 Seconds

We mentioned before that an aggregation period of 10 seconds with vehicles running at 100 km/h could lead to many vehicles entering a road with an accident due to the fact that vehicles could travel 278 m in that period of time. Even worse, it could take several aggregation periods, until a consensus is reached by the vehicles about the traffic condition and a traffic incident report is sent. This situation is shown to be particularly critical for the 1750 and 2000 systems. To test this hypothesis we reduced the aggregation period to 2 seconds and we present the results in the following sections. Not all the scenarios will be considered due to the time consuming nature of these simulations, and the 8000 System is introduced and evaluated.

7.3.2.1 The 500 System with Aggregation Period of 2 Seconds. In Table 8 the results from the simulation for the 500 System are presented, for both the rebroadcasting and non-rebroadcasting algorithms, with an aggregation period of 2 seconds. In Fig. 19 these results are presented in graphical form.

Table 8. 500 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 2 seconds

System	500 No Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	802.19	0.00
0.05	686.13	14.47
0.10	696.02	13.23
0.15	721.53	10.05
0.20	685.71	14.52
0.25	676.92	15.62
0.30	660.92	17.61
0.35	657.50	18.04
0.40	713.17	11.10
0.45	666.47	16.92
0.50	671.99	16.23
0.55	674.68	15.90
0.60	666.55	16.91
0.65	654.89	18.36
0.70	678.70	15.39
0.75	695.77	13.27
0.80	666.86	16.87
0.85	636.96	20.60
0.90	737.84	8.02
0.95	700.52	12.67
1.00	606.61	24.38

System	500 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	802.19	0.00
0.05	551.68	31.23
0.10	569.55	29.00
0.15	579.77	27.73
0.20	648.18	19.20
0.25	571.18	28.80
0.30	592.73	26.11
0.35	557.48	30.50
0.40	552.99	31.06
0.45	561.06	30.06
0.50	579.46	27.76
0.55	586.88	26.84
0.60	569.96	28.95
0.65	540.50	32.62
0.70	528.67	34.10
0.75	541.68	32.47
0.80	539.99	32.68
0.85	540.23	32.66
0.90	494.24	38.39
0.95	461.90	42.42
1.00	524.57	34.61

It can be seen in Fig. 19, that there is a clear improvement in traffic when rebroadcasting is used with a peak improvement of 40% for an evaporation factor of 90%. The non-rebroadcasting case becomes less sensitive to evaporation, but does not show a significant improvement with respect to the case with aggregation period of 10 seconds.

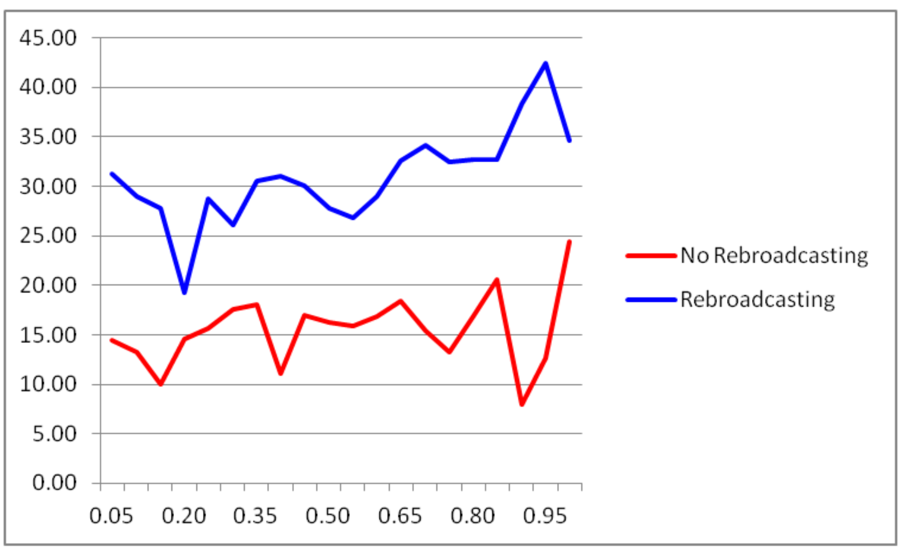


Fig. 19. 500 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 10 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.2.2 The 1750 System with Aggregation Period of 2 Seconds. In Table 9 we show the results from the simulation for the 1,750 System, for both the rebroadcasting and non-rebroadcasting algorithms, with an aggregation period of 2 seconds. In Fig. 20 these results are presented in graphical form.

Table 9. 1750 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 2 seconds

System	1750 No Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	911.23	0.00
0.05	905.00	0.68
0.10	905.00	0.68
0.15	906.41	0.53
0.20	907.22	0.44
0.25	906.94	0.47
0.30	909.84	0.15
0.35	908.11	0.34
0.40	907.19	0.44
0.45	906.40	0.53
0.50	907.30	0.43
0.55	908.03	0.35
0.60	909.76	0.16
0.65	**906.70**	0.50
0.70	**910.36**	0.10
0.75	907.19	0.44
0.80	910.35	0.10
0.85	904.86	0.70
0.90	910.46	0.08
0.95	915.17	-0.43
1.00	954.21	-4.72

System	1750 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	911.23	0.00
0.05	732.35	19.63
0.10	725.86	20.34
0.15	743.46	18.41
0.20	752.95	17.37
0.25	702.00	22.96
0.30	722.44	20.72
0.35	769.88	15.51
0.40	716.95	21.32
0.45	709.72	22.11
0.50	690.25	24.25
0.55	748.42	17.87
0.60	687.00	24.61
0.65	**659.50**	**27.62**
0.70	**665.13**	**27.01**
0.75	645.30	29.18
0.80	616.07	32.39
0.85	661.35	27.42
0.90	588.02	35.47
0.95	544.26	40.27
1.00	534.87	41.30

It can be seen in Fig. 20, that the non-rebroadcasting algorithm is completely useless to improve traffic in this system, an anticipated result derived from the limited range of the VANET transceivers. We would like to highlight here, the fact that, for the first time, an evaporation factor with value of 1 would increase the average travel time. This observation can be explained as follows: an evaporation factor of 1 indicates that the traffic conditions stored in the memory of the vehicles become static, or in other words, persistent. As the accident information is not being able to efficiently leave the accident edge, it cannot be used by upcoming vehicles. However, some transient events would be permanently stored in memory, as if they were a long lasting accident, and lead to wrong route selections. On the other hand, there is a huge improvement on the rebroadcasting algorithm, which is useless under the 10 second aggregation period and now presents improvements of up to 40%. In the following sections we will analyze the 4000 System and the 8000 System under rebroadcasting algorithm only, as it is shown here that the non-rebroadcasting algorithm does not work in large systems.

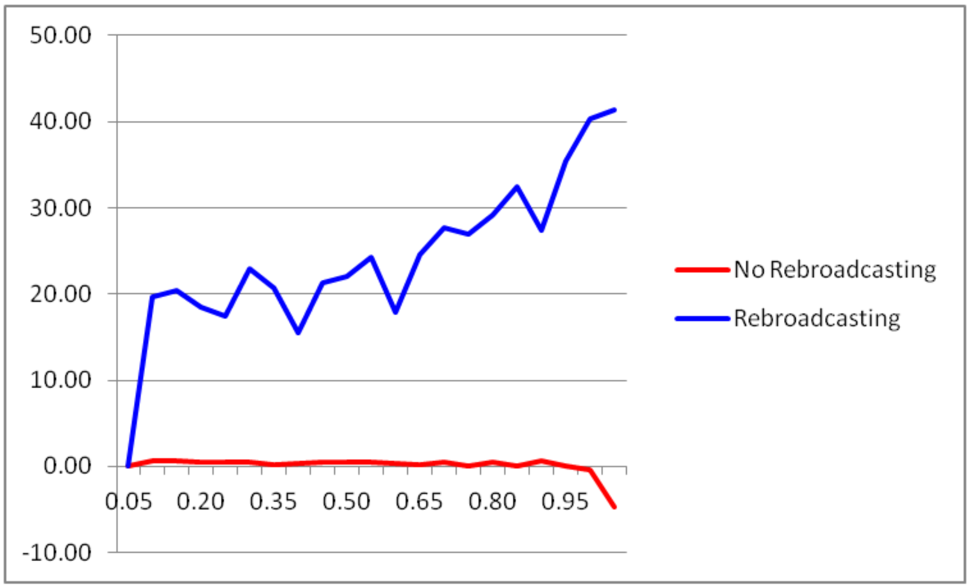


Fig. 20. 1750 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 2 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.2.3 The 4000 System with Aggregation Period of 2 Seconds. In Table 10. we show the results from the simulation for the 4000 System, for the rebroadcasting algorithm only, with an aggregation period of 2 seconds. In Fig. 21 these results are presented in graphical form. It can be seen in Fig. 21, that this aggregation period provides a consistent improvement of the average travel time for all the evaporation factors values. The peak gain also improved from around 17 to 22. However, we start to see that the gains are not as profound as those of the smaller systems. In the next section we show that for the 8000 System there is virtually no gain. Similar to the 1750 System, we can explain this new limit by the fact that as the length of the road increases, the number of vehicles needed to propagate the message out of the road segment is close to the number of vehicles trapped in that segment when to algorithm is used.

Table 10. 4000 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 2 seconds

System	4000 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	1269.55	0.00
0.05	1133.94	10.68
0.10	1175.27	7.43
0.15	1183.82	6.75
0.20	1176.18	7.35
0.25	1139.90	10.21
0.30	1171.99	7.68
0.35	1221.68	3.77
0.40	1109.73	12.59
0.45	1188.83	6.36
0.50	1134.40	10.65
0.55	1149.30	9.47
0.60	1159.31	8.68
0.65	1168.23	7.98
0.70	1131.81	10.85
0.75	1134.10	10.67
0.80	1131.26	10.89
0.85	1058.01	16.66
0.90	1059.46	16.55
0.95	993.35	21.76
1.00	1004.03	20.91

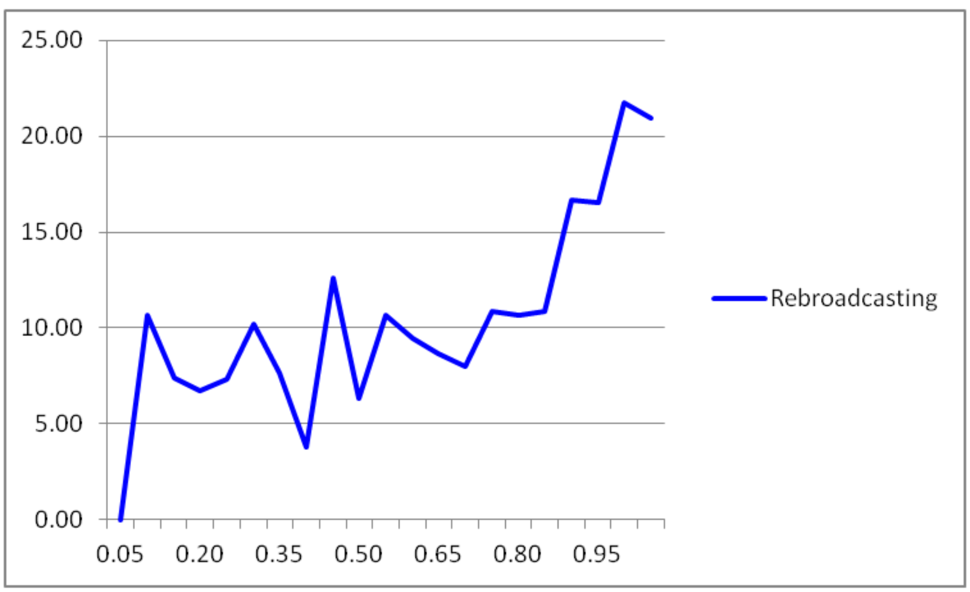


Fig. 21. 4000 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 2 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

7.3.2.4 The 8000 System with Aggregation Period of 2 Seconds. In Table 11 we show the results from the simulation for the 8000 System, for the rebroadcasting algorithm only, with an aggregation period of 2 seconds. In Fig. 22 these results are presented in graphical form. It can be seen in Fig. 22, that there is no gain on the average travel time in this system.

Table 11. 8000 System, rebroadcasting and non-rebroadcasting algorithm data for ap = 2 seconds

System	8000 Rebroadcasting	
Evaporation Factor	Average Travel Time (s)	Improvement %
NA	1766.29	0.00
0.05	1769.45	-0.18
0.10	1742.98	1.32
0.15	1769.77	-0.20
0.20	1733.31	1.87
0.25	1773.14	-0.39
0.30	1763.69	0.15
0.35	1779.92	-0.77
0.40	1767.56	-0.07
0.45	1751.37	0.84
0.50	1742.59	1.34
0.55	1771.46	-0.29
0.60	1772.17	-0.33
0.65	1765.53	0.04
0.70	1770.36	-0.23
0.75	1770.52	-0.24
0.80	1756.06	0.58
0.85	1759.95	0.36
0.90	1769.30	-0.17
0.95	1740.02	1.49
1.00	1746.43	1.12

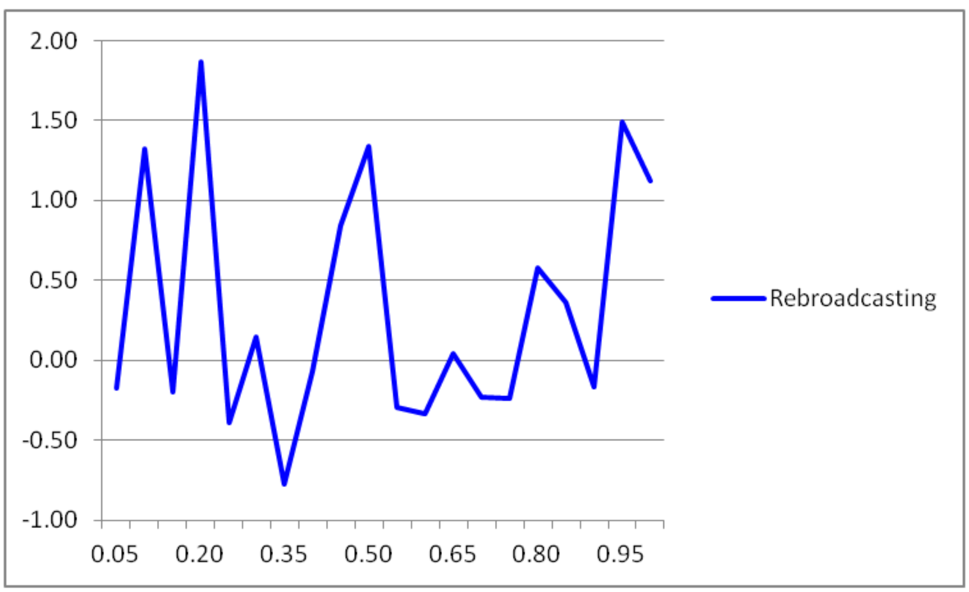


Fig. 22. 8000 System, comparisson of rebroadcasting and non-rebroadcasting algorithm for AP = 2 Seconds. Horizontal Axis: Evaporation Factor. Vertical Axis: Average Travel Time Improvement %.

8. CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Simulation of Vehicular ad hoc networks is a complex process necessary for testing algorithms in this environment. We introduce and evaluate Road-ACO, a novel decentralized algorithm to alleviate traffic congestion on road networks and to fill the void left by current algorithms which are either static, centralized, or require infrastructure. Road-ACO is an algorithm inspired by Ant Colony Optimization for the Dynamic Traffic Assignment in VANETS. Initial results indicate a promising future for approaches based on this algorithm. Simulation results for the algorithm show an improvement on the average travel time of 29.17%, over the SR case when one of the two segments leading to the destination has heavier traffic demand and less favorable lane connections. It is important to indicate that Road-ACO is a realistic approach as it improves traffic as it evolves, in real time, without prior knowledge of the traffic demand or the schedule of the cars that will enter the road network in the future. Also Road-ACO enjoys the benefits of being decentralized and infrastructure-less. We evaluate the algorithm in road networks with segments exceeding 1,000 m to test the algorithm in systems where rebroadcasting is needed. We observe that the rebroadcasting version of the algorithm performs better when we use short aggregation periods. We determine that the aggregation period is an important factor: if it is too long, vehicles are not informed in a timely manner and the algorithm does not perform well. For an aggregation period of 2 seconds we observe significant improvement of 20% for road segments of 4,000 m, while for the case of 1,750 m we observe 40%. Our investigation considers vehicles travelling in one direction of the road system and we anticipate further improvements, both in efficiency and road segments' lengths when traffic is introduced in the other direction and vehicles travelling this direction are allowed to use a store-carry-and-forward mechanism.

8.2 Future Work

8.2.1 Joint Optimization of Traffic Assignment and Traffic Lights

Traffic light cycles have an important impact on traffic. By extending the algorithm to control the traffic lights we expect it to achieve further improvement on the average *travel time*.

8.2.2 Evaporation as a Function of Traffic Light Cycle, Edge Length, and accident severity

Traditional ACO algorithms use evaporation to handle good paths that decay over time. Our algorithm uses evaporation to handle bad paths that improve over time. We believe that the evaporation factor is dependent on the particular road characteristics. We propose to evaluate the influence of the road segment length and the traffic light cycle on this parameter. We consider that it may be interesting to evaluate the use of several types or colors of pheromones concurrently, one type could be used to optimize no accident conditions, another color could be used for mild accidents, and a third color for serious accidents. We expect a different evaporation factor in each case will be required.

9. REFERENCES

[1] "Our Nation's Highways - onh.pdf." [Online]. Available: http://www.fhwa.dot.gov/policyinformation/pubs/pl08021/pdf/onh.pdf. [Accessed: 24-Oct-2013].

[2] N. Grace, C. Oxley, S. Sloan, A. Tallon, P. Thornton, T. Black, and A. V. Easton, "Transforming Transportation through Connectivity: ITS Strategic Research Plan, 2010–2014 (Progress Update, 2012)," 2012. [Online]. Available: http://trid.trb.org/view.aspx?id=1218083. [Accessed: 24-Oct-2013].

[3] D. Schrank, B. Eisele, and T. Lomax, "TTI's 2012 Urban Mobility Report Powered by INRIX Traffic Data," *Texas A&M Transportation Institute. The Texas A&M University System*, 2012.

[4] J. G. Wardrop, "ROAD PAPER. SOME THEORETICAL ASPECTS OF ROAD TRAFFIC RESEARCH.," in *ICE Proceedings: Engineering Divisions*, 1952, vol. 1, pp. 325–362.

[5] M. Beckmann, C. B. McGuire, and C. B. Winsten, "Studies in the Economics of Transportation," 1956.

[6] S. Yagar, "Dynamic traffic assignment by individual path minimization and queuing," *Transportation Research*, vol. 5, no. 3, pp. 179–196, 1971.

[7] W. Y. Szeto and S. C. Wong, "Dynamic traffic assignment: model classifications and recent advances in travel choice principles," *Central European Journal of Engineering*, vol. 2, no. 1, pp. 1–18, 2012.

[8] S. Peeta and A. K. Ziliaskopoulos, "Foundations of dynamic traffic assignment: The past, the present and the future," *Networks and Spatial Economics*, vol. 1, no. 3–4, pp. 233–265, 2001.

[9] C. F. Daganzo and Y. Sheffi, "On stochastic models of traffic assignment," *Transportation Science*, vol. 11, no. 3, pp. 253–274, 1977.

[10] R. B. Dial, "A probabilistic multipath traffic assignment model which obviates path enumeration," *Transportation research*, vol. 5, no. 2, pp. 83–111, 1971.

[11] H. von Falkenhausen, "Traffic assignment by a stochastic model," in *Proceedings of the Fourth International Conference on Operational Research*, 1966, pp. 415–421.

[12] C. F. Daganzo, "Stochastic Network Equilibrium with Multiple Vehicle Types and Asymmetric, Indefinite Link Cost Jacobians," *Transportation Science*, vol. 17, no. 3, pp. 282–300, Aug. 1983.

[13] S. Maerivoet and B. De Moor, "Transportation planning and traffic flow models," *arXiv preprint physics/0507127*, 2005.

[14] Y.-C. Chiu, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks, "Dynamic traffic assignment: A primer," *Transportation Research E-Circular*, no. E-C153, 2011.

[15] M. Carey and E. Subrahmanian, “An approach to modelling time-varying flows on congested networks,” *Transportation Research Part B: Methodological*, vol. 34, no. 3, pp. 157–183, 2000.
[16] M. Carey, “Nonconvexity of the dynamic traffic assignment problem,” *Transportation Research Part B: Methodological*, vol. 26, no. 2, pp. 127–133, 1992.
[17] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses, “Selfish routing in capacitated networks,” *Mathematics of Operations Research*, vol. 29, no. 4, pp. 961–976, 2004.
[18] G. E. Cantarella, “A General Fixed-Point Approach to Multimode Multi-User Equilibrium Assignment with Elastic Demand,” *Transportation Science*, vol. 31, no. 2, pp. 107–128, May 1997.
[19] M. Dorigo and T. Stützle, *Ant Colony Optimization*, First ed. MIT Press, 2004.
[20] W. Arellano and I. Mahgoub, “Swarm Intelligence Inspired Routing Algorithms for Ad Hoc Wireless Networks,” in *Encyclopedia of Wireless and Mobile Communications, Second Edition*, Taylor & Francis, 2013, pp. 1720–1734.
[21] M. Dorigo, “Optimization, learning and natural algorithms,” *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.
[22] M. Dorigo, V. Maniezzo, A. Colorni, M. Dorigo, M. Dorigo, V. Maniezzo, V. Maniezzo, A. Colorni, and A. Colorni, “Positive Feedback as a Search Strategy,” Dipartimento di Elettronica - Politecnico di Milano, Milano, Italy, Report 91-016, 1991.
[23] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[24] J. L. Deneubourg, J. M. Pasteels, and J. C. Verhaeghe, “Probabilistic behaviour in ants: A strategy of errors?,” *Journal of Theoretical Biology*, vol. 105, no. 2, pp. 259–271, 1983.
[25] J.-L. Deneubourg and S. Goss, “Collective patterns and decision-making,” *Ethology Ecology & Evolution*, vol. 1, no. 4, pp. 295–311, 1989.
[26] S. Goss, R. Beckers, J.-L. Deneubourg, S. Aron, and J. M. Pasteels, “How trail laying and trail following can solve foraging problems for ant colonies,” in *Behavioural Mechanisms of Food Selection*, Springer, 1990, pp. 661–678.
[27] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
[28] M. López-Ibáñez and C. Blum, “Beam-ACO for the travelling salesman problem with time windows,” *Computers & Operations Research*, vol. 37, no. 9, pp. 1570–1583, Sep. 2010.
[29] L.-N. Xing, Y.-W. Chen, P. Wang, Q.-S. Zhao, and J. Xiong, “A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems,” *Applied Soft Computing*, vol. 10, no. 3, pp. 888–896, Jun. 2010.
[30] J. Heinonen and F. Pettersson, “Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem,” *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 989–998, Apr. 2007.

[31] F. J. Martínez, F. González-Vidosa, A. Hospitaler, and V. Yepes, "Heuristic optimization of RC bridge piers with rectangular hollow sections," *Computers & Structures*, vol. 88, no. 5–6, pp. 375–386, Mar. 2010.
[32] Z. Yu, O. C. Au, R. Zou, W. Yu, and J. Tian, "An adaptive unsupervised approach toward pixel clustering and color image segmentation," *Pattern Recognition,* vol. 43, no. 5, pp. 1889–1906, May 2010.
[33] J. Cai, X. Ma, Q. Li, L. Li, and H. Peng, "A multi-objective chaotic ant swarm optimization for environmental/economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 5, pp. 337–344, Jun. 2010.
[34] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Arxiv preprint arXiv:1105.5449*, 2011.
[35] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.
[36] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical Computer Science*, vol. 344, no. 2–3, pp. 243–278, Nov. 2005.
[37] B. Chandra Mohan and R. Baskaran, "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618–4627, Mar. 2012.
[38] L. Mussone, M. Ponzi, and M. Matteucci, "Ant colony optimization for stochastic user equilibrium," *I sistemi stradali di trasporto nella società dell'informazione: monitoraggio, simulazione e predisposizione di basi informative dinamiche*, pp. 175–199, 2005.
[39] L. D'Acierno, B. Montella, and F. D. Lucia, "A Stochastic Traffic Assignment Algorithm Based on Ant Colony Optimisation," in *Ant Colony Optimization and Swarm Intelligence*, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds. Springer Berlin Heidelberg, 2006, pp. 25–36.
[40] M. Matteucci and L. Mussone, "Ant Colony Optimization Technique for Equilibrium Assignment in Congested Transportation Networks," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA, 2006, pp. 87–88.
[41] L. D'Acierno, M. Gallo, and B. Montella, "An Ant Colony Optimisation algorithm for solving the asymmetric traffic assignment problem," *European Journal of Operational Research*, vol. 217, no. 2, pp. 459–469, Mar. 2012.
[42] Z. Cong, B. De Schutter, and R. Babuška, "Ant Colony Routing algorithm for freeway networks," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 1–19, 2013.
[43] J. E. Bell and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Advanced Engineering Informatics*, vol. 18, no. 1, pp. 41–48, Jan. 2004.
[44] R. Claes, T. Holvoet, and D. Weyns, "A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 364–373, Jun. 2011.
[45] M. R. J. Sattari, H. Malakooti, A. Jalooli, and R. M. Noor, "A Dynamic Vehicular Traffic Control Using Ant Colony and Traffic Light Optimization," in *Advances in*

Systems Science, J. Swiątek, A. Grzech, P. Swiątek, and J. M. Tomczak, Eds. Springer International Publishing, 2014, pp. 57–66.

[46] P. Bedi, N. Mediratta, S. Dhand, R. Sharma, and A. Singhal, "Avoiding Traffic Jam Using Ant Colony Optimization - A Novel Approach," in *International Conference on Conference on Computational Intelligence and Multimedia Applications, 2007*, 2007, vol. 1, pp. 61–67.

[47] B. Tatomir and L. Rothkrantz, "Hierarchical Routing in Traffic Using Swarm-Intelligence," in *IEEE Intelligent Transportation Systems Conference, 2006. ITSC '06*, 2006, pp. 230–235.

[48] W. Narzt, U. Wilflingseder, G. Pomberger, D. Kolb, and H. Hortner, "Self-organising congestion evasion strategies using ant-based pheromones," *Intelligent Transport Systems, IET*, vol. 4, no. 1, pp. 93–102, 2010.

[49] B. Tatomir, L. J. Rothkrantz, and A.-C. Suson, "Travel time prediction for dynamic routing using ant based control," in *Simulation Conference (WSC), Proceedings of the 2009 Winter*, 2009, pp. 1069–1078.

[50] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[51] D. K. Merchant and G. L. Nemhauser, "A Model and an Algorithm for the Dynamic Traffic Assignment Problems," *Transportation Science*, vol. 12, no. 3, pp. 183–199, Aug. 1978.

[52] M. Matteucci and L. Mussone, "An ant colony system for transportation user equilibrium analysis in congested networks," *Swarm Intell*, vol. 7, no. 4, pp. 255–277, Dec. 2013.

[53] J. G. Kohl, *Der Verkerhr und die Ansiedelungen der Menschen in inhrer abhängigkeit von der Gestaltung der Erdoberfläche..* Arnold, 1850.

[54] H. Kawashima, "Japanese perspective of driver information systems," *Transportation*, vol. 17, no. 3, pp. 263–284, 1990.

[55] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[56] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (VANETs): challenges and perspectives," in *ITS Telecommunications Proceedings, 2006 6th International Conference on*, 2006, pp. 761–766.

[57] W. Arellano and I. Mahgoub, "TrafficModeler extensions: A case for rapid VANET simulation using OMNET++, SUMO, and VEINS," in *Proceedings of the IEEE 10th international symposium on High Capacity Optical Networks and Enabling Technologies (HONET 2013)*, Magosa, 2013, pp. 109–115.

[58] L. G. Papaleondiou and M. D. Dikaiakos, "Trafficmodeler: A graphical tool for programming microscopic traffic simulators through high-level abstractions," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, 2009, pp. 1–5.

[59] "SUMO - Simulation of Urban Mobility." [Online]. Available: http://sumo.sourceforge.net/. [Accessed: 06-Apr-2013].

[60] "OpenStreetMap." [Online]. Available: http://www.openstreetmap.org/. [Accessed: 13-Apr-2013].

[61] "Veins." [Online]. Available: http://veins.car2x.org/. [Accessed: 30-Mar-2014].

[62] W. Arellano, I. Mahgoub, and M. Ilyas, "Veins extensions to implement a message based algorithm for Dynamic Traffic Assignment in VANETs simulations," in *Proceedings of the IEEE 11th international symposium on High Capacity Optical Networks and Enabling Technologies (HONET 2014)*, Charlotte, NC, 2014, pp. 29–35.

[63] W. Arellano and I. Mahgoub, "Reverse Online Algorithm for the Dynamic Traffic Assignment inspired by Ant Colony Optimization for VANETs," *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 3, no. 111, p. 2, 2014.

[64] M. G. McNally, "The four step model," Institute of Transportation Studies - University of California, Irvine, CA, UCI-ITS-WP-07-2, 2007.

[65] M. G. McNally and C. Rindt, "The activity-based approach," Institute of Transportation Studies - University of California, Irvine, CA, UCI-ITS-WP-07-1, 2007.

[66] R. B. Mitchell and C. Rapkin, *Urban Traffic–A Function of Land Use*. New York, NY: Columbia University Press, 1954.

[67] E. Weiner, *Urban transportation planning in the United States: An historical overview*. Greenwood Publishing Group, 1999.

[68] M. M. Rahman, S. Saha, U. Chengan, and A. S. Alfa, "IP traffic matrix estimation methods: comparisons and improvements," in *Communications, 2006. IEEE International Conference on*, 2006, vol. 1, pp. 90–96.

[69] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *ACM SIGCOMM Computer Communication Review*, 2002, vol. 32, pp. 161–174.

[70] J. Nash, "Non-cooperative games," *The Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[71] P. M. Jones, M. C. Dix, M. I. Clarke, and I. G. Heggie, *Understanding travel behaviour*. Brookfield, VT: Gower Publishing, 1983.

[72] K. W. Axhausen, "Activity-based modelling: Research directions and possibilities," *New Look at Multi-Modal Modelling. Department of Environment, Transport and the Regions, London, Cambridge and Oxford*, 2000.

[73] R. J. Smeed, "Some circumstances in which vehicles will reach their destinations earlier by starting later," *Transportation Science*, vol. 1, no. 4, pp. 308–317, 1967.

[74] V. D. S. Eleuterio, "Static Traffic Assignment Problem. A comparison between Beckmann (1956) and Nesterov & de Palma (1998) models," in *Conference paper STRC 2007*, Monte Veritá, Switzerland, 2007.

[75] A. De Palma and I. E. Nesterov, "Optimization formulations and static equilibrium in congested transportation networks," Université Catholique de Louvain, Louvain-la-Neuve, Belgium, CORE discussion paper 9861, 1998.

[76] M. Bergomi, "Traffic Assignment Problem: The Stochastic User Equilibrium," Master Thesis, Department of Mathematics, ETH, Zurich, Switzerland, 2009.

[77] A. S. Leeuwen, "Static Traffic Assignment with Queuing," Master Thesis Applied Mathematics, University of Twente, Enschede, Netherlands, 2011.

[78] S. Bradley, A. Hax, and T. Magnanti, *Applied mathematical programming*. Addison-Wesley, 1977.

[79] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, 1951, vol. 5.

[80] D. Boyce, “Beckmann’s transportation network equilibrium model: Its history and relationship to the Kuhn–Tucker conditions,” *Economics of Transportation*, vol. 2, no. 1, pp. 47–52, Mar. 2013.

[81] *Traffic assignment manual for application with a large, high speed computer*. Washington,: U.S. Department of Commerce - Bureau of Public Roads, 1964.

[82] Y. Nesterov and A. de Palma, “Stationary dynamic solutions in congested transportation networks: summary and perspectives,” *Networks and spatial economics*, vol. 3, no. 3, pp. 371–395, 2003.

[83] M. Frank and P. Wolfe, “An algorithm for quadratic programming,” *Naval research logistics quarterly*, vol. 3, no. 1–2, pp. 95–110, 1956.

[84] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla, “An efficient approach to solving the road network equilibrium traffic assignment problem,” *Transportation Research*, vol. 9, no. 5, pp. 309–318, 1975.

[85] H. Bar-Gera, “Origin-based algorithms for Transportation Network Modeling,” National Institute of Statistical Sciences, Research Triangle Park, NC, Report 103, Nov. 1999.

[86] M. Jeter, *Mathematical programming: an introduction to optimization*, vol. 102. CRC press, 1986.

[87] A. Nagurney, “Variational Inequality Theory,” presented at the Management Science Seminar, Amherst MA, 2014.

[88] S. Dafermos, “Traffic Equilibrium and Variational Inequalities,” *Transportation Science*, vol. 14, no. 1, pp. 42–54, Feb. 1980.

[89] A. Nagurney, *Network Economics: A Variational Inequality Approach*. Springer Science & Business Media, 1998.

[90] A. Suson, “Dynamic routing using ant-based control,” Master thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2010.

[91] E. J. Schmitt and H. Jula, “Vehicle Route Guidance Systems: Classification and Comparison,” in *IEEE Intelligent Transportation Systems Conference, 2006. ITSC ’06*, 2006, pp. 242–247.

[92] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[93] W. B. Powell and Y. Sheffi, “The Convergence of Equilibrium Algorithms with Predetermined Step Sizes,” *Transportation Science*, vol. 16, no. 1, pp. 45–55, Feb. 1982.

[94] S. P. Hoogendoorn and P. H. Bovy, “State-of-the-art of vehicular traffic flow modelling,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, no. 4, pp. 283–303, 2001.

[95] S. Algers, E. Bernauer, M. Boero, L. Breheret, C. Di Taranto, M. Dougherty, K. Fox, and J.-F. Gabard, *Review of micro-simulation models*. Institute for Transport Studies, 1997.

[96] G. F. Newell, “A simplified car-following theory: a lower order model,” *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195–205, Mar. 2002.
[97] D. C. Gazis, R. Herman, and R. W. Rothery, “Nonlinear follow-the-leader models of traffic flow,” *Operations Research*, vol. 9, no. 4, pp. 545–567, 1961.
[98] P. G. Gipps, “A behavioural car-following model for computer simulation,” *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.
[99] S. Krau, “Towards a unified view of microscopic traffic flow theories,” *IFAC Transportation Systems*, vol. 2, p. 941, 1997.
[100] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO-Simulation of Urban MObility-an Overview,” in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, 2011, pp. 55–60.
[101] M. Rahman, M. Chowdhury, Y. Xie, and Y. He, “Review of Microscopic Lane-Changing Models and Future Research Opportunities,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1942–1956, Dec. 2013.
[102] P. Jesus, C. Baquero, and P. S. Almeida, “A Survey of Distributed Data Aggregation Algorithms,” *Arxiv preprint arXiv:1110.0725*, 2011.
[103] K. Ibrahim and M. C. Weigle, “Cascade: Cluster-Based Accurate Syntactic Compression of Aggregated Data in Vanets,” in *GLOBECOM Workshops, 2008 IEEE*, 2008, pp. 1–10.
[104] I. Salhi, M. O. Cherif, and S. M. Senouci, “A New Architecture for Data Collection in Vehicular Networks,” in *Proceedings of the 2009 IEEE International Conference on Communications*, 2009, pp. 1–6.
[105] A. Peinado, A. Ortiz, and J. Munilla, “Secure Distributed System inspired by Ant Colonies for Road Traffic Management in Emergency Situations,” in *SECURWARE 2013, The Seventh International Conference on Emerging Security Information, Systems and Technologies*, 2013, pp. 144–149.
[106] W. S. HOMBURGER, *Traffic Estimation: Computer Programs for Educational Purposes (Course Notes)*. Berkeley, CA: Institute of Transportation and Traffic Engineering. University of California, 1969.
[107] B. G. Heydecker and J. D. Addison, “An exact expression of dynamic traffic equilibrium,” *Transportation and traffic theory*, pp. 359–383, 1996.
[108] M. Carey, “Optimal Time-Varying Flows on Congested Networks,” *Operations Research*, vol. 35, no. 1, pp. 58–69, Jan. 1987.
[109] T. L. Friesz, J. Luque, R. L. Tobin, and B.-W. Wie, “Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem,” *Operations Research*, vol. 37, no. 6, pp. 893–901, 1989.
[110] B.-W. Wie, “Dynamic Analysis of User-Optimized Network Flows with Elastic Travel Demand,” *Transportation Research Record*, no. 1328, 1991.
[111] B. Ran, D. E. Boyce, and L. J. LeBlanc, “A New Class of Instantaneous Dynamic User-Optimal Traffic Assignment Models,” *Operations Research*, vol. 41, no. 1, pp. 192–202, Feb. 1993.
[112] T. L. Friesz, D. Bernstein, T. E. Smith, R. L. Tobin, and B. W. Wie, “A Variational Inequality Formulation of the Dynamic Network User Equilibrium Problem,” *Operations Research*, vol. 41, no. 1, pp. 179–191, Feb. 1993.

[113] W. B. Arthur, “Inductive Reasoning and Bounded Rationality,” *The American Economic Review*, vol. 84, no. 2, pp. 406–411, May 1994.
[114] S. Peeta, “System Optimal Dynamic Traffic Assignment in Congested Networks with Advanced Information Systems,” Dissertation, University of Texas at Austin, 1994.
[115] N. H. Gartner, “Optimal Traffic Assignment with Elastic Demands: A Review Part II. Algorithmic Approaches,” *Transportation Science*, vol. 14, no. 2, pp. 192–208, May 1980.
[116] K. Nagel and C. L. Barrett, “Using microsimulation feedback for trip adaptation for realistic traffic in Dallas,” *International Journal of Modern Physics C*, vol. 8, no. 3, pp. 505–525, 1997.
[117] S. Fischer and B. Vöcking, “Adaptive Routing with Stale Information,” in *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, 2005, pp. 276–283.
[118] Y. Hollander and J. N. Prashker, “The applicability of non-cooperative game theory in transport analysis,” *Transportation*, vol. 33, no. 5, pp. 481–496, Sep. 2006.
[119] O. Chen and M. Ben-Akiva, “Game-Theoretic Formulations of Interaction Between Dynamic Traffic Control and Dynamic Traffic Assignment,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1617, no. 1, pp. 179–188, Jan. 1998.
[120] C.-H. Chou and J.-C. Teng, “A fuzzy logic controller for traffic junction signals,” *Information Sciences*, vol. 143, no. 1–4, pp. 73–97, Jun. 2002.
[121] J. C. Spall and D. C. Chin, “Traffic-responsive signal timing for system-wide traffic control,” *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 3–4, pp. 153–163, Aug. 1997.
[122] S. Zhu, D. M. Levinson, and L. Zhang, “An Agent-Based Route Choice Model,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 1743621, Apr. 2007.
[123] P. Mirchandani, R. Syal, and D. Lucas, “Traffic Assignment Using Iterated Routebased Simulation,” in *Proc. of 82st Annual Meeting of the Transportation Research Board (CD-ROM)*, 2003.
[124] Y. Pavlis and M. Papageorgiou, “Simple Decentralized Feedback Strategies for Route Guidance in Traffic Networks,” *Transportation Science*, vol. 33, no. 3, pp. 264–278, Aug. 1999.
[125] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, “Network state estimation and prediction for real-time traffic management,” *Networks and Spatial Economics*, vol. 1, no. 3–4, pp. 293–318, 2001.
[126] M. Ben-Akiva, M. Bierlaire, J. Bottom, H. Koutsopoulos, and R. Mishalani, “Development Of A Route Guidance Generation System For Real-Time Application,” in *Proceedings of the IFAC Transportation Systems 97 Conference*, Chania, 1997.
[127] R. Kroon and L. Rothkrantz, “Dynamic vehicle routing using an ABC-algorithm,” *Collected papers on the PITA project*, vol. 21, 2003.
[128] G. Di Caro and M. Dorigo, “AntNet: Distributed stigmergetic control for communications networks,” *VIVEK-BOMBAY-*, vol. 12, no. 3/4, pp. 2–37, 1999.

[129] Q. Meng, H. Yang, and M. G. H. Bell, "An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem," *Transportation Research Part B: Methodological*, vol. 35, no. 1, pp. 83–105, Jan. 2001.
[130] O. Baskan and S. Haldenbilen, "Ant Colony Optimization Approach for Optimizing Traffic Signal Timings," in *Ant Colony Optimization Methods And Applications*, 2011, p. 205.
[131] O. Baskan, S. Haldenbilen, H. Ceylan, and H. Ceylan, "A new solution algorithm for improving performance of ant colony optimization," *Applied Mathematics and Computation*, vol. 211, no. 1, pp. 75–84, May 2009.
[132] B. Tatomir and L. J. M. Rothkrantz, "H-ABC: A scalable dynamic routing algorithm," *Recent Advances in Artificial Life*, vol. 5, p. 8, 2005.
[133] C. J. Eyckelhof and M. Snoek, "Ant systems for a dynamic TSP," in *Ant Algorithms*, Springer, 2002, pp. 88–99.
[134] "Kendall Corridor Transportation Alternatives Analysis (Kendall-Link) Final Report, September 2007 - MPO_kendalllink_final_200709.pdf." [Online]. Available: http://www.miamidade.gov/mpo/docs/MPO_kendalllink_final_200709.pdf. [Accessed: 16-Apr-2013].
[135] A. Varga, "OMNeT++ Home Page," 1997. [Online]. Available: http://www.omnetpp.org/.