DOCTORAL THESIS

Natural Computing for Vehicular Networks

Author:
Jamal TOUTOUH

Supervisor:
Dr. Enrique ALBA

PhD Thesis Dissertation in Computer Science

Department of *Lenguajes y Ciencias de la Computación*

Escuela Técnica Superior de Ingeniería Informática

UNIVERSIDAD DE MÁLAGA

December, 2015

AUTOR: Jamal Toutouh El Alamin

http://orcid.org/0000-0003-1152-0346

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Carta de aval

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

El Dr. **Enrique Alba**, Catedrático de Universidad perteneciente al Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga,

certifica

que D. **Jamal Toutouh**, Ingeniero en Informática por la Universidad de Málaga, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

Natural Computing for Vehicular Networks

Revisado el presente trabajo, estimo que puede ser presentado al tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en la legislación vigente, autorizo la presentación de la Tesis Doctoral en la Universidad de Málaga.

En Málaga, 13 de noviembre de 2015

Firmado: Dr. Enrique Alba Torres

Acknowledgements

A PhD Thesis is a long and arduous process, which has only been possible thanks to the help, support, and contribution that I received from many people.

I would like to thank my supervisor, Prof. Enrique Alba, for encouraging me to perform this work, for guiding me, and most importantly, for his patience. To the NEO Research Group members, not only to the current people in the team, but also to some of them who have already left and the visitors who have stayed with us just for short periods of time. I have learned uncountable new things during our discussions, meetings, and coffee breaks. I have found here more than colleagues, we are a great family (Briseida, Christian, Daniel, David, Francis, Gabriel, Guillermo, Hajer, Javier Apolloni, Javier Ferrer, Juanjo, Martín Bilbao, Martín Pedemonte, Pablo, Paco, Raúl, Sofiane, Yesnier, among many others). Additionally, I want to thank Sergio Nesmachnow and his endless energy for stimulating me.

I cannot forget to make mention of Joni Jämsä, Dr. Fethi Filali, and Dr. Zeeshan Hameed Mir, among others, who supported me and facilitated my stays at Centria University of Applied Sciences in Finland and at QMIC in Qatar.

Of course, I feel grateful to my family. Ahmed, my father, who one day, when he was still a young man, decided to leave his small hamlet in Morocco (Beni Sidel) to provide his family with a brighter future. Sakina, my mother, who is the strongest woman that I have ever met. She fought very hard to give us a great education, even when my father had passed away. The most important heritage that my parents have left us is the importance of the hard work and being honest. Mohamed and Said, my brothers, have always been there for me when I needed them. Finally, I want to highlight another person. I remember him taking care of me since I was a little child. He was always my best friend, my support, a model to follow, and he was with me in many adventures. He is my eldest brother, Abdeslam.

Also, thanks to Laila, Petete, and Dikra for convincing/forcing me to leave Melilla to pursue a dream, when I was bombarded with doubts about my ability to study Computer Science at university.

Also, thanks to my collages/friends of Engineering Without Borders (Ingeniería Sin Fronteras, Málaga), who I bored to death complaining about my hard work, for giving me advices even without understanding a word of what I was saying. Specially, I am grateful to Carmina, Marta, and Paco who supported me during some very complicated moments.

To my friends (football and volleyball teammates), Anto, Marioli, Pablo Antón, Charlie, Cristobal, Guillermo, Sergio, etc., I'm grateful for the numerous good moments we shared during this time and I hope we can have many others in a near future.

And last but of course not the least, I want to thank Rosa for staying on my side every day, and giving me her immeasurable support and unconditional encouragement. She shared with me all those bad moments of these intense years and she made me feel that they were not that bad. Furthermore, she was the right person to be with during the good times. Thank you very much Rosa.

Financial support:

Jamal Toutouh was partially supported by FPU AP2010-3108 Grant of the Spanish Ministry of Education. Also, this PhD thesis work has been partially funded by the Spanish MINECO and FEDER project TIN2014-57341-R (http://moveon.lcc.uma.es).

Contents

List of Abbreviations

ABC	**A**rtificial **B**ee **C**olony
ACO	**A**nt **C**olony **O**ptimization
AHVN	**A**dvanced **H**eterogeneous **V**ehicular **N**etworks
AIS	**A**rtificial **I**mmune **S**ystems
AMR	**A**daptive **M**essage **R**outing
AU	**A**pplication **U**nit
BER	**B**it **E**rror **R**ate
BEH	**B**alloon **E**xpansion **H**euristic
BIP	**B**inary **I**nteger **P**rogramming
BMR	**B**eacon **M**odification **R**equest
BR	**B**eacon **R**ate
BRAC	**B**eacon **R**ate **A**daptation **C**omponent
CASS	**C**ooperative **A**ctive **S**afety **S**ystem
CO	**C**hannel **O**ccupancy
CSMA	**C**arrier-**S**ense **M**ultiple **A**ccess
CVS	**C**ooperative **V**ehicle **S**afety
DE	**D**ifferential **E**volution
DBR	**D**esirable **B**eacon **R**ate
DFCN	**D**elayed **F**looding **C**umulative **N**eighborhood
DRP	**D**ata **R**e**P**ly
DRQ	**D**ata **R**e**Q**uest
DSRC	**D**irect **S**hort **R**ange **C**ommunications
DYMO	**DY**namic **M**ANET **O**n-demand
EA	**E**volutionary **A**lgorithm
EP	**E**volutionary **P**rogramming
ERR	**E**ffective **R**adio **R**ange
ES	**E**volution **S**trategies
FBR	**F**air **B**eacon **R**ate
FIRP	**F**ile **I**nformation **R**e**P**ly
FIRQ	**F**ile **I**nformation **R**e**Q**uest
FREEDY	**F**air beacon **R**ate gr**EEDY**
FTC	**F**ile **T**ransfer **C**onfiguration
GA	**G**enetic **A**lgorithm
GD	**G**enerational **D**istance
genGA	**gen**erational **G**enetic **A**lgorithm
GSM	**G**lobal **S**ystem for **M**obile
GP	**G**enetic **P**rogramming

GPS	**G**lobal **P**ositioning **S**ystem
HMI	**H**uman **M**achine **I**nterface
HVN	**H**ybrid **V**ehicular **N**etwork
HyBR	**Hy**brid **Bee** swarm **R**outing
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
ICT	**I**nformation and **C**ommunication **T**echnologies
IGRP	**I**ntersection-based **G**eographical **R**outing **P**rotocol
ITS	**I**ntelligent **T**ransport **S**ystems
LTE	**L**ong **T**erm **E**volution
MCP	**M**aximum **C**overage **P**roblem
MCTTP	**M**aximum **C**overage with **T**ime **T**hreshold **P**roblem
MID	**M**ultiple **I**nterface **D**eclaration
MO	**M**ulti-**O**bjective
MOEA	**M**ulti-**O**bjective **E**volutionary **A**lgorithm
MOP	**M**ulti-objective **O**ptimization **P**roblem
MPR	**M**ulti**P**oint **R**elay
NC	**N**atural **C**omputing
NIC	**N**etwork **I**nterface **C**ard
NSGA-II	**N**on-dominated **S**orting **G**enetic **A**lgorithm-II
OBD	**O**n-**B**oard **D**iagnosis
OBU	**O**n-**B**oard **U**nit
pEA	**p**arallel **E**volutionary **A**lgorithm
PSO	**P**article **S**warm **O**ptimization
QoS	**Q**uality-**o**f-**S**ervice
QoSBeeVANET	**Q**uality **o**f **S**ervice **Bee** Swarm routing protocol for **VANET**
RAT	**R**adio **A**ccess **T**echnologie
RND	**R**adio **N**etwork **D**esign
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
RSU	**R**oad**S**ide **U**nit
RSU-DP	**R**oad**S**ide **U**nit **D**eployment **P**roblem
SA	**S**imulated **A**nnealing
SIEC	**S**warm **I**nformation **E**xchange **C**omponent
SMPSO	**S**peed-constrained **M**ulti-objective **P**article **S**warm **O**ptimization
SQMC	**S**elf **Q**ueue **M**onitoring **C**omponent
ssGA	**s**teady **s**tate **G**enetic **A**lgorithm
SUMO	**S**imulation of **U**rban **MO**bility
TC	**T**opology **C**ontrol
TS	**Tabu** **S**earch
V2B	**V**ehicle-to-**B**roadband
V2I	**V**ehicle-to-**I**nfrastructure
V2V	**V**ehicle-to-**V**ehicle
VACO	**V**ehicular routing protocol based on **A**nt **C**olony **O**ptimization
VANET	**V**ehicular **A**d hoc **NET**works
VDTP	**V**ehicular **D**ata **T**ransfer **P**rotocol
VITP	**V**ehicular **I**nformation **T**ransfer **P**rotocol
WAVE	**W**ireless **A**ccess in **V**ehicular **E**nvironment
WLAN	**W**ireless **L**ocal **A**rea **N**etworks
WME	**W**AVE **M**anagement **E**ntity

Chapter 1

Introduction

THIS volume is a technical summary of the work done in the previous years technically aimed at incorporating new intelligent algorithms to the design of vehicular networks. The scientific hypothesis has been that of proving that Natural Computing, and specially some metaheuristics, can lead to actual contributions to the domain of communicating vehicles in hostile environments (high interference, intense mobility, low computational resources, etc.).

This PhD thesis does so in addressing both, academic and industrial challenges in the intelligent algorithms side, as well as in the modern technologies needed to this end. After a wide set of original proposals, complex simulations, and real tests in the city, we conclude that our initial hypothesis does hold: bio-inspired optimization algorithms and related tools are a net contribution to upgrade existing services in communicating with cars as well as they open new lines of research, development, and innovation of international interest.

1.1 Motivation

The vast use of private cars causes today serious road traffic problems that have to be solved in some way by our modern society. Recent estimates have shown that each year more than 1.2 million human lives pass away as the result of road traffic accidents. Besides this, traffic jams bother the daily life of the population mainly because they cause longer trip times and larger associated pollution, not to mention the economic losses due to the delays and other transport issues. At last, the longstanding promise of deploying applications to improve efficiency and safety in road transport is becoming a reality. A number of *smart mobility* solutions based on *intelligent transport systems* (ITS) are hitting the market, e.g., Waze (Waze, 2009) or TomTom (TomTom, 2003).

The main idea behind this technology consists in sharing with road users information about the traffic conditions by using wireless communication (via FM radio broadcast or, more recently, via cellular networks). A better informed driver may make better decisions about a road journey, since the misjudgment of drivers is the major cause of accidents and traffic jams. Even, a given intelligent system could automatically take these decisions more efficiently, and then, inform the driver. A plethora of other applications for passengers is also possible because of vehicular communication networks. However, there are several drawbacks to most of these previous services: **i)** they are centralized and based on a fixed and costly infrastructure, e.g., over-roadway and in-roadway sensors; **ii)** such systems provide traffic information only about the main roads in the city, otherwise, it could entail huge costs of installation; and **iii)** the information updates are in the range of 20-50 minutes, far from ideal real-time awareness.

To overcome such drawbacks, *vehicular ad hoc networks* (VANETs) emerge as spontaneous networks of self organized vehicles and roadside infrastructure elements, which continually exchange data with each other by using short range wireless communications. VANETs provide the possibility of using vehicles and roadside elements as sources of information (by using their internal sensors) to monitor the actual (updated) traffic conditions. But also, they strive to harness the power of ubiquitous communication for continuously keeping the vicinity of vehicles aware about the current/future maneuvers. Thus, drivers/vehicles may anticipate hazardous events and avoid wrong driving decisions.

The major stress in this new type of modern ad hoc networks is put onto providing this ubiquitous computing technology without using any central manager entity. VANETs must be capable to exchange information in real time, using the wireless medium efficiently, adapting the communication protocols to the continuous changes in road traffic environments, etc.

VANETs have a set of hard constraints that need to be handled to obtain the mentioned features. This results in new open challenges that can be formulated as unsolved (and hard to solve) optimization problems, which cannot be efficiently tackled by using classic (exact) optimization techniques. These problems present novel models, one or more opposing objectives, limited computational resources, and constrained network capabilities. These computational and network resources must be taken into account in two different terms: first, the optimization process to address the problems has to be performed in a constrained time and by using a limited computation platform; and second, the computed solutions must handle restricted resources (wireless medium, energy, economic cost, etc.), in addition to the need of being accurate and robust. As a consequence, there is a need of modern, new, and very advanced methods to address all these challenges.

Bio-inspired inspired algorithms are a widely used *Natural Computing* (NC) methods for tackling complex (hard-to-solve and NP) optimization problems. These algorithms gradually perform improvements to near-optimal solutions in a timely fashion. NC in optimization is becoming more and more popular within the research community. Some of the main reasons for this are: **1)** they present a reduced computational complexity, **2)** they are generic tools (do not require full problem knowledge), and **3)** they can be tailored in terms of numerical accuracy, memory size, etc. All these features motivated us to apply NC to address optimization problems in the domain of vehicular networks. Thus, we establish as the main goal of this thesis dissertation to show the feasibility of using NC for solving VANET optimization problems, as well as to produce useful configurations of real software and hardware for VANETs. Accordingly, we aim at identifying some relevant open optimization challenges in this domain, we define their formulation, we engineer a set of NC algorithms to tackle them, we show their effectiveness through experimental evaluation (including statistical validation), and finally, we test some of the obtained solutions throughout outdoor experiments (*in vitro*) by using real vehicles and wireless devices.

As it can be seen, this thesis has a strong focus towards the applicability of the results obtained, and this is the reason that this research work has been developed in connection to several research projects aiming at deploying real world VANETs, such as *EUREKA-CELTIC CARLINK* (CARLINK, 2006), *DIRICOM* (DIRICOM, 2008), *roadME* (roadME, 2011), *CoMoSeF* (CoMoSeF, 2012), *CellCar* (CellCar, 2013), and *MAXCT* (MAXCT, 2014).

1.2 Objectives and Phases

This thesis focuses on the resolution of complex optimization problems in the domain of VANETs by using NC. This general goal can be detailed into the following concrete objectives:

- Identify the most important challenges that arise in the emerging field of vehicular networks. Select a subset of these problems to be tackled as optimization problems in this thesis dissertation.
- Propose a formulation for each of the optimization problems selected.
- Describe the NC algorithms that will be used to address the problems.
- Define general optimization methodologies based in NC that lead to the resolution of the specific problems addressed, but that can be used in new further optimization problems in VANETs.
- Design novel strategies that enhance the performance of current NC techniques, either from the perspective of the quality of the solutions produced, or from the perspective of the computational effort required to reach them.
- Demonstrate the effectiveness of the NC based optimization tools and the solutions computed by statistically assessed experimental evaluation.
- Validate the solutions computed in real world pilots.

For the research develop in this thesis, we follow the phases of the Scientific Method (Dodig-Crnkovic, 2002). The first phase is *observation*, where we study the innovative field of VANETs to detect the main challenges and target problems. We identify data dissemination and roadside unit platform design as two groups of relevant problems, where it is possible to make contributions. After that, we formulate the *question* about applying NC to efficiently address the observed open issues in VANETs, which can be expressed as hard-to-solve optimization problems. The general *hypothesis* of this dissertation is formulated as "*NC is suitable to efficiently address such VANET optimization problems, overcoming some limitations of traditional methods*". Afterwards, we perform the *experimentation* phase. We review the literature about VANET optimization problems, and proposed new variants. These problems are solved by using different NC methods in order to evaluate their performance against the current state-of-the-art methods. Experimenting with new technologies for communications is also a target of the research. The next phase of the scientific method is the *analysis* of the results. We perform an in-depth analysis for each target problem, in order to study different metrics to evaluate the behavior of the proposed NC algorithms. Statistical tests are applied to validate the confidence of the results. Finally, we draw the *conclusions* according to the experience extracted from the research process. As the main conclusion, the obtained results support our working hypothesis about the applicability of NC to efficiently solve relevant VANET optimization problems.

1.3 Thesis Contributions

The main contribution of this PhD thesis is the in depth analysis of the use of NC to address specific vehicular communication problems, being our proposals in many cases the first approaches found in the literature (to the best knowledge) in this domain. This contribution is summarized as the following outputs:

- A review of the state of the art in VANETs, paying special attention to routing data, broadcasting beacons, and roadside unit deployment issues. Additionally, a literature analysis of the different NC is performed.

- An in depth study in off-line protocol optimization in VANETs. Different formulations have been proposed according to the type of protocol (routing or file transfer) and the main purpose of optimization (QoS and/or energy efficiency). Additionally, we have also analyzed mono-objective and multi-objective models of these problems. Therewith, we have defined a general off-line optimization tool, which couples a NC method (e.g., an evolutionary algorithm or a swarm intelligence method) with a VANET simulator, that can be tailored to optimize any VANET protocol.

- An improvement of the efficiency and the efficacy of the NC algorithms used to solve off-line optimization in VANETs by introducing parallel models and specific operators (e.g., initialization, mutation, and crossover).

- A new formulation of the broadcasting optimization problem more in line with the specific features and requirements of the VANET's has been defined. In this sense, four different distributed dynamic broadcasting algorithms have been proposed to address such optimization problem.

- A novel multi-objective optimization problem formulation of the roadside unit deployment in vehicular communications (RSU-DP problem), which takes into account real-wold existing devices and road traffic information. In order to address this problem, a large-sized instance with the main roads of Málaga (Spain) has been modeled with real traffic data from the city council. Additionally, we have proposed a bio-inspired multi-objective algorithm (with specific operators) to tackle RSU-DP in the city of Málaga. A comparison with the last state-of-the-art algorithms has been provided.

- A large VANET simulation testbed defined based on real-world data of different neighborhoods of Málaga has been created. The instances that incorporate this testbed comprise different area sizes, road traffic patterns, VANET applications, etc.

- A definition of a real-world outdoor testbed to evaluate different VANET solutions. Besides this, we have analyzed the feasibility of expanding VANET technology with the use of lightweight personal devices (smartphones, tablets, and laptops).

1.4 Dissertation Outline

This thesis work is highly oriented towards the problem domain, and this reflects into its structure as a document. Thus, this volume is divided into three parts, that follow this introduction. In the *first part* we present the fundamentals and bases for the work: the vehicular networks domain, NC as a global family of methods to solve optimization problems, and the VANET optimization problems addressed in this work, including the literature review. The *second part* (the principal one) is devoted to the whole experimental research work performed in this thesis. Full analysis of the optimization problems are analyzed here (routing, broadcasting, and roadside infrastructure design). As most of them are pioneers in the current literature, full formulation, models, and evaluation methods are detailed for each problem. Additionally, this part includes the real-wold VANET experiments performed on the roads of Málaga. Finally, the *third and last part* of the thesis regroups the main conclusions drawn throughout the work and summarizes our findings. We detail the contents of the chapters below.

- **Part I: Fundamentals**

  Chapter 2 gives a general description of VANETs. It introduces some of the most important communication technologies, reviews the main VANET applications, provides the main differences between VANETs and other mobile networks, and overviews the different worldwide VANET-related projects and consortia. Finally, we discuss the main current open challenges, which have been the main motivation to formulate the optimization problems that are presented in the next chapter.

  Chapter 3 gives an introduction to the research field of NC in optimization problems. We put special attention on the algorithms utilized in the whole thesis. Besides this, it introduces the VANET optimization problems analyzed and reviews the main related literature. Finally, this chapter includes the methodology followed to evaluate the optimization techniques applied in solving such problems.

- **Part II: Optimization and Experimentation in Vehicular Networks**

  Chapter 4 presents our approach to solve off-line protocol tuning optimization problems for VANETs. Different VANET protocols (file transfer and routing) are optimized in terms of quality-of-service (QoS) and/or energy consumption. For this reason, several problem formulations are presented for the problem objectives. In our analysis, we include a representative set of well-known NC algorithms, which include mono-objective and multi-objective solvers and sequential (one-thread) and parallel algorithms. Finally, the efficacy of the optimized protocols are confirmed by performing a set of validation experiments, which involve simulations over a number of realistic VANET scenarios.

  In Chapter 5, we introduce the concept of fair (balanced) beacon broadcasting. Then, we describe a series of distributed greedy algorithms for broadcasting beacons in VANET. These methods, that are supposed to be running in each vehicle, dynamically (on-line) optimize the balance and the VANET medium occupancy. Finally, a set of validation experiments are performed in a number of highway VANET scenarios.

  In Chapter 6, we present a novel explicit multi-objective formulation of the RSU-DP in order to find a set of solutions that maximize the QoS and minimize the installation costs. We devise a parallel evolutionary algorithm, which applies specific operators to improve the efficacy of this optimization method, to solve the problem in an instance defined with real data of Málaga. Finally, the results are compared against the last state-of-the-art optimization algorithms proposed to address RSU-DP.

  Chapter 7 describes a set of real-world VANET experiments that have been performed in open roads of the city of Málaga. These outdoor testbeds are performed with two main purposes: first, we compare the optimized and the standard parameterizations of a file transfer protocol to confirm the results obtained in Chapter 4; and second, we analyze real vehicle-to-vehicle communications by using a set of widely available personal portable devices (smartphones, tablets, and laptops) in order to evaluate their capabilities to be used to deploy real VANETs.

- **Part VI: Conclusions and Future Work**

  Chapter 8 contains a global review of this thesis dissertation, and regroups the main conclusions drawn for the whole research work. The thesis objectives and main contributions are discussed in view of the results obtained. Finally, the future lines of research that can be pursued following the work presented here are briefly sketched and discussed.

In summary, we have tried to give a new perspective to significant problems (in the domain of VANETs), we have applied new solving techniques focusing on the applicability of the solution, and we have analyzed how to make vehicular networks a reality. We have performed innovative research to fill gaps of the literature in order to conclude a meaningful PhD thesis in this competitive present world.

Part I: Fundamentals

Any fool can know. The point is to understand.
Albert Einstein

Chapter 2

Vehicular Networks: Opportunities and Research Challenges

TODAY, the longstanding promise of deploying applications based on *vehicular ad hoc networks* (VANETs) to improve efficiency and safety in road transport is becoming a reality. However, there are still some open questions to be carefully addressed before a widespread deployment of this technology. The aim of this chapter is to introduce VANETs as a case of success of the implementation of Natural Computing as a design tool. Thus, this chapter introduces some of the most important VANET communication technologies, it reviews the most salient applications that rely on these type of networks, it presents the main differences between VANETs and other mobile networks, it overviews the different activities (projects and consortia) carried out in this domain, and finally, it describes the most salient current open challenges. Please notice that it does not intend to be a comprehensive analysis of the state-of-the-art of vehicular communications since it is out of the scope of this thesis.

2.1 Introduction

Nowadays, the most widely used means of transport are cars and other private vehicles. The huge increase in the volume of road traffic experienced during the last decades causes today serious problems that have to be confronted by our modern society. Recent estimates have shown that 1.24 million human lives are lost each year as a result of road traffic crashes (WHO, 2015). In addition, traffic jams, besides causing discomfort, limit the efficiency in transportation with vehicles because of the growth of the travel times, with the subsequent increase of the energy consumption and the impact on the environment (affected by the large associate air pollution).

As a global consequence, our present industry, academia, and governments worldwide are devoting considerable resources to increase *road safety* and *traffic efficiency*. *Information and communication technologies* (ICT) are the driving force behind some of the most important innovations in dealing with this challenge. The use of mobile communication systems in vehicular environments is a recent field that will provide modern *intelligent transportation systems* (ITS) that will allow the development of the next generation smart mobility services. One of the most promising solutions is the use of emerging vehicular communication networks known as *vehicular ad hoc networks* (VANETs) (Hartenstein and Laberteaux, 2009; Campolo et al., 2015; Yousefi et al., 2006; Morris et al., 2000). VANETs are composed by a collection of vehicles and roadside elements connected with each other using infrastructure-less wireless technologies. Although, many considerations are still in the air, like the open debate whether it is necessary or not to redefine the term in order to include infrastructure-based and cellular systems in the vehicular communications (see Figure 2.1). VANETs offer the possibility of improving the safety and efficiency of the road traffic through powerful cooperative applications based on a continuous information sharing.

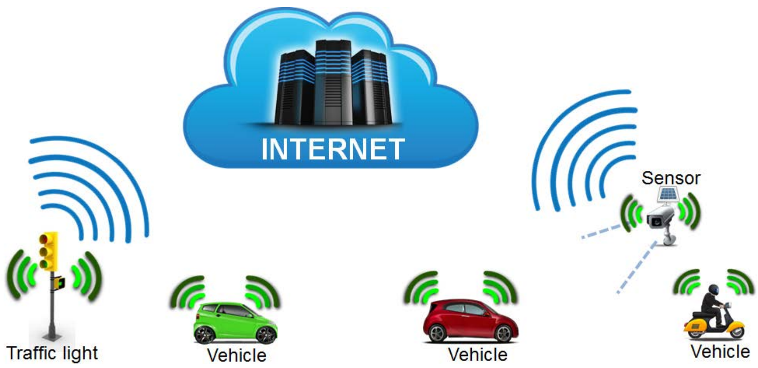


FIGURE 2.1: A schematic representation of a vehicular network.

2.2 VANET Communication Technologies

ITS applications have different and varied communication requirements in terms of bandwidth, latency, coverage and other performance metrics, which have to be fulfilled by VANETs. This section introduces the VANET's architecture, the different communication domains, and the main radio access technologies utilized in these vehicular environments.

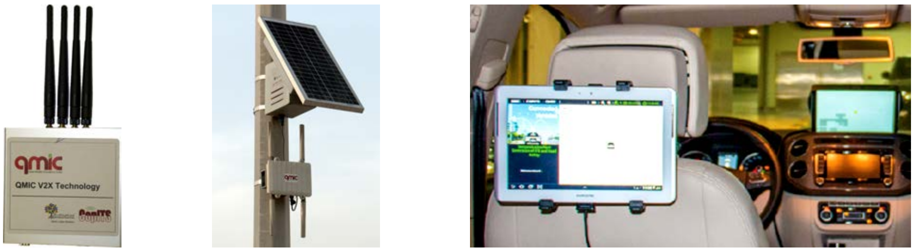
2.2.1 Architecture

The three main components of the whole VANET system are the on-board, roadside, and application units. They are described bellow:

- **On-board unit (OBU):** OBUs are hardware devices integrated in the vehicles in order to provide them with processing and communication capabilities. Their main functions are: **i)** gathering and processing the data collected from the sensors installed in the vehicle (e.g., kinematics data) and **ii)** exchanging vehicular information with other VANET nodes (OBUs or roadside units) via *direct short range wireless communications* (DSRC), which are principally based on IEEE 802.11p radio technology (FCC, 1999). OBUs may additionally include another network interfaces based on other radio access technologies such IEEE 802.11a/b/g/n or WiMAX (IEEE 802.16). They also provide communication services to the application units forwarding data on behalf of other OBUs in the network (see Figure 2.2.a).

- **Roadside unit (RSU):** RSUs are devices that are usually installed on the roadside infrastructure elements, i.e., traffic lights or signals. In turn, they may be fixed along roadside as specific dedicated VANET elements. They include a network interface to exchange information with other VANET nodes through DSRC. They may also be equipped with other network interfaces to connect to other networks or to the Internet (see Figure 2.2.b). They perform three main functions (C2C-CC, 2007): **i)** acting as an information source or receiver in safety applications, e.g., warning about of the existence of roadworks; **ii)** extending the effective communication range by forwarding data to other VANET nodes (OBUs or RSUs) through multi-hop communications; and **iii)** providing Internet connectivity to the OBUs.

- **Application unit (AU):** The AU may be either an external device connected to the OBU or a device integrated into the OBU forming a single physical unit (see Figure 2.2.c). If it is external device, it can be a dedicated device for VANET applications or a regular mobile device, such as a smartphone or a *personal digital assistant* (PDA). The AU provides a user interface to access the VANET applications, e.g, microphone or speakers. It is connected to the OBU through wired or wireless connection, such as Bluetooth. The OBUs act as network gateway that is used by the AUs to exchange information through the VANET or to connect to the Internet.



a) OBU. b) RSU. c) AU.

FIGURE 2.2: Main elements of the VANET architecture (QMIC, 2015).

2.2.2 Communication Domains

Vehicular communication systems include of three types of communication domains (Olariu and Weigle, 2009):

- **In-vehicle domain**: This domain refers to the network composed by the devices inside the vehicle (i.e, the sensors, the OBU, and one or multiple AUs). Thus, the OBU is able to use the network links to gather the information from the sensors and share it with the AUs. The communication can be implemented with wired or wireless links. Some example of the used wireless technologies in this domain are the Bluetooth, the *wireless universal serial bus* (WUSB), and the *ultra-wide band* (UWB) (Jogi and Choudhary, 2009).

- **Ad hoc domain**: The ad hoc domain is composed by OBUs (mobile devices) and RSUs forming a *mobile ad hoc network* (MANET). These nodes exchange information in a fully distributed manner without using any centralized coordination entity through DSRC. In this domain, the communicating nodes are either just vehicles (OBUs) exchanging information with each other through ***vehicle-to-vehicle*** **(V2V)** communication or vehicles communicating with RSUs via ***vehicle-to-infrastructure*** **(V2I)** communication (see Figure 2.3). The performance of the communications in the ad hoc domain is highly dependent on the routing and broadcasting protocols utilized to forward the data from the source to the destination nodes.

- **Infrastructural domain**: Extending VANETs by including links to Internet provider agents, may increase the power of the VANET applications. Thus, RSUs can be attached to this kind of networks in order to act as gateway for OBUs allowing the vehicles to access services that are provided by infrastructure networks. In turn, OBUs may be equipped with cellular radio network interfaces, such as *long term evolution* (LTE), to perform ***vehicle-to-broadband cloud*** **(V2B)** communication to directly access infrastructure networks (see Figure 2.3) (Hossain et al., 2010).

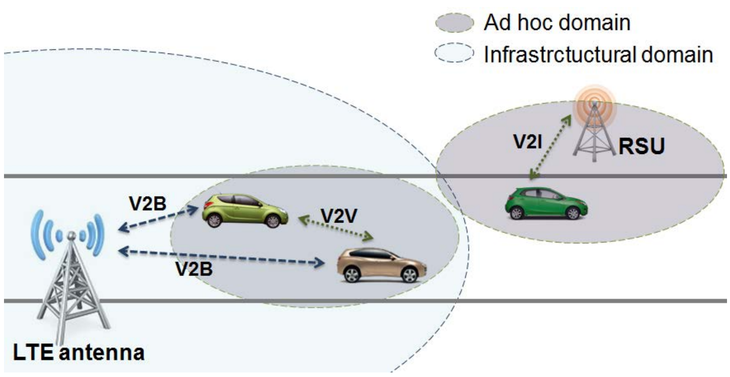


FIGURE 2.3: A representation of the ad hoc and the infrastructural communication domains.

In spite of the raising of different radio access technologies of cellular networks that may provide infrastructural domain vehicular communications (V2B), most research is addressed in proposing an ad hoc communication platform to allow reliable V2V and V2I communications (Hartenstein and Laberteaux, 2009).

2.2.3 Wireless Access Technologies for VANETs

Vehicular networks can be deployed by using several of the numerous access technologies available today (Hossain et al., 2010). The *radio access technologies* (RATs) applied in vehicular environments, such as wireless local area networks (WLAN based on IEEE 802.11a/b/g/n/p standards), WiMAX (IEEE 802.16 a/e standards), Bluetooth (IEEE 802.15.1 standard), and last generations of cellular networks (3G and LTE), provide communications with different quality-of-service (QoS) which determine the communication latency, bandwidth, coverage, etc. The most salient RATs applied in V2V, V2I, and V2B communications are the following ones:

- **WLAN**: The family of IEEE 802.11 wireless standards (Cooklev, 2004), which have achieved a great acceptance in the market, support short-range relatively high-speed data transmission. IEEE 802.11 standard defines over-the-air protocols necessary to support networking in a local area and it specifies *physical* (PHY) and *medium access control* (MAC) layers. There are several specifications in the IEEE 802.11 family which extend the original one (IEEE 802.11). The most extended ones are the IEEE 802.11b and IEEE 802.11g standards, that provide 11 Mbps and 54 Mbps transmission in the 2.4 GHz band with a maximum range of 500 m, respectively. IEEE 802.11a is an extension to 802.11 that provides up to 54 Mbps in the 5 GHz band using the *orthogonal frequency division multiplexing* (OFDM) encoding scheme. Transfer rates increased with IEEE 802.11n standard with a bandwidth up to 500 Mbps. In addition, there are a number of other 802.11 WG activities that define inter access point protocol (IEEE 802.11f), MAC enhancements for security (IEEE 802.11i), MAC enhancements for QoS (IEEE 802.11e), etc. WLAN radio access technologies enable V2V and V2I communications.

- **DSRC**: In 1999, the *U.S. Federal Communication Commission* allocated 75 MHz of DSRC spectrum at 5.9 GHz to be used exclusively for vehicular ad hoc communications (FCC, 1999). DSRC technology allows high speed communications between VANET nodes that might be separated up to 1000 meters. There exist differences in the frequency allocation between North America and Europe, but the intention is to be able to use the same antenna and transmitter/receiver. Different organizations like the *Institute of Electrical and Electronic Engineers* (IEEE), *International Standard Organization* (ISO) or *Car-to-Car Communication Consortium* (C2C-CC) are working on developing an architecture for VANETs.

There is no agreement between the different organizations on which of the different proposals is more convenient for vehicular networks, thus, each of them is working on their own proposal: *wireless access in vehicular environment* (WAVE) by IEEE (Uzcategui and Acosta-Marum, 2009), *communication access for land mobiles* (CALM) by ISO (CALM, 2015), and *Car-to-Car Network* (C2CNet) by C2C-CC (C2C-CC, 2007).

Nowadays, the most utilized architecture to provide DSRC (V2V and V2I communications) is based on the IEEE 802.11p standard (ETSI, 2010), which is specifically designed for supporting WAVE ITS applications. The IEEE 802.11p was firstly adopted by IEEE and *American Society for Testing and Materials* (ASTM) (ASTM, 2003) and specifies the PHY and MAC layers. WAVE protocol stack is completed by the IEEE 1609 standards family (Guerrero-Ibáñez et al., 2013) at the upper layers.

- **WiMAX**: WiMax is a standard developed by IEEE, the IEEE 802.16 (Nuaymi, 2007). It was defined to provide wireless broadband access over long distances (up to 50 Km). WiMAX can operate with bandwidths up to 70 Mbps. However, when the distance between the nodes grows the *bit error rate* (BER) critically increases, and therefore, the effective bandwidth is reduced (Ghosh et al., 2005). In 2005, Mobile WiMAX (originally based on 802.16e-2005) was deployed in many countries to provide mobile nodes with WiMAX radio access. WiMAX has been proposed to improve the performance of V2I communications (Shrivastava et al., 2012).

- **LTE**: *Long term evolution* (LTE), also known as *Evolved Universal Terrestrial Access Network* (E-UTRAN), is a standard for cellular wireless communication that has been developed by the *Third Generation Partnership Project* (3GPP) 3GPP, 2010. The main idea that motivated LTE was to improve the *Universal Mobile Telecommunications System* (UMTS), which is the third generation (3G) of mobile cellular systems for networks based on the *Global System for Mobile* (GSM) standard. LTE provides high spectral efficiency by a combination of advanced multi-antenna techniques, and *orthogonal frequency-division multiple access* (OFDMA) in the *downlink* (DL) and *single-carrier frequency-division multiple access* (SC-FDMA) in the *uplink* (UL). Thus, it improves significantly the data rates, with the potential for 300 Mbps downstream and 75 Mbps upstream, while reducing the communication latency. In addition, it offers scalable bandwidth capacity and backwards compatibility with existing GSM and UMTS technologies. Some authors have included this RAT to be used in vehicular communications because some of these features are ideal for ITS applications (Benslimane et al., 2011; Mosyagin, 2010).

Since ITS applications request services with different communication requirements in terms of latency, bandwidth, error rate, coverage area, etc., recently, several studies urged to combine different RATs into a unified *hybrid vehicular networking* (HVN) or ***V2X-Communication*** architecture (Hameed Mir and Filali, 2014; Park et al., 2014; Vinel, 2012). For example, ISO proposed CALM M5 by incorporating a set of wireless technologies including UMTS-3G, infrared communication, and wireless systems adapted to IEEE 802.11p (Olariu and Weigle, 2009).

2.3 VANET Applications

Vehicular networks allow the development of a large set of powerful applications that will improve the road transportation experience for both, drivers and passengers. Typically, the associated literature categorizes these applications in two different main groups: ***safety*** and ***non-safety*** applications. The first ones utilize VANETs to exchange information to improve road safety and avoid road accidents. Non-safety applications represent a hodgepodge of different applications that include the enhancement of the traffic efficiency and the improvement of the passengers' comfort and entertainment, among others. Nevertheless, these groups cannot be seen completely orthogonal. Thus, an application designed to prevent car crashes and hazardous situations improves the efficiency because it avoids the traffic jams that the accidents may cause. Following, we summarize the most salient traffic safety, efficiency, and comfort applications and their main network QoS requirements.

2.3.1 VANET Safety Applications

Road accidents may involve loss of human lives, injuries, cost of hospitalization, and damage to the property and vehicles. These are some of the reasons that urge governmental institutions to thoughtfully address road safety. Thus, future VANET safety related applications may help to reduce the number of hazardous situations that road users may suffer.

These applications gather real-time information from diverse sources (through vehicle's sensors, received from other nodes or both), process it, and disseminate it to the other nodes in the form of safety messages or *beacons*. Most of these applications rely on periodic message broadcasting, also known as *beaconing*. The frequency of these messages depends on the nature of the application and ranges from one to tens of hertz depending on the update frequency of the required information. These applications can be classified into five main categories (Al-Sultan et al., 2014):

1. *Cooperative driving*: Continuously sharing information with/from other vehicles about kinematics (direction, velocity or acceleration), future maneuvers, etc. reduces the uncertainty about the behavior of other vehicles, and therefore, reduces the probability of hazardous situations. Thus, cooperative driving applications aim at improving safety by exchanging information with the other VANET nodes through V2V and V2I communications. The information used by these applications can be gathered by the OBU through the sensors and received from other nodes in the neighborhood. Designing applications and protocols that support these kind of applications receive a lot interest from the research community and industry (Van Arem et al., 2006; Santamaria et al., 2015). This category includes a great number of possible applications, some of those are: *cooperative collision warning*, *road condition warning*, *emergency electronic brake lights*, *lane change warning*, *pre-crash sensing*, *highway merge warning*, and *cooperative adaptive cruise control* (*vehicle platooning*).

2. *Intersection collision avoidance*: An important number of hazardous situations occur at crossroads and intersections. Therefore, improving the classic intersection collisions system (traffic lights and signs) will help preventing many road accidents. These applications are based on the idea of gathering information about the vehicles close to the intersection through road sensors and OBUs. Thus, if the system infers that there is a possibility of existence of a dangerous situation or an accident, a warning message is sent to the OBUs in order to alert the drivers so that they can take the appropriate actions to avoid it. Some examples of these applications are (Al-Sultan et al., 2014): *intersection collision warning*, *warn about violating stop sign*, *left turn assistant* (see Figure 2.4.a), and *pedestrian crossing*.

3. *Public safety*: The main tasks of public safety applications are: **i)** to warn other drivers about that an accident has occurred to avoid new dangerous situations, and **ii)**, to support emergency teams by minimizing their travel time and by providing information about the accident situation. Two proposed applications in this category are: *post crash warning* and *approaching emergency vehicle warning* (see Figure 2.4.b).

4. *Vehicle diagnostics and maintenance*: This category of applications aim to evaluate the state of the vehicle, to notify the driver about safety defects, and to remind inattentive drivers about the time to go for vehicle maintenance. In addition, they allow drivers to download personalized vehicle settings better adapted to their driving profile. Some examples of vehicle diagnostics and maintenance applications are: *safety recall notice* and *just-in-time repair notification*.

5. *Sign extension*: These applications improve classic roadside traffic signs by advertising the approaching vehicles in order to reduce the potential danger due to inattention of the drivers. Also, sign extension applications may adapt the provided information to the driver profile or to some specific situation. For example, the speed limit signs may further restrict the speed during rainy times. *Work zone warning* (see Figure 2.4.c), *wrong way driver warning*, *curve speed warning*, and *low bridge warning*, among others, are included in sign extension applications.

2.3.2 VANET Traffic Efficiency Applications

Most of the classical ITS services deployed by authorities and institutions regarding traffic improvement are based on two main ideas: **i)** analyzing the information gathered by installed sensors on fixed spots on the road network and **ii)** informing the driver about the traffic status and possible route alternatives via roadside information panels, or publishing this information periodically on the mass media (radio or television).

However, the core idea of VANET traffic efficiency applications is to use vehicles as mobile sensors which monitor useful traffic data such as traffic density, road weather conditions and parking occupancy. Thus, vehicles automatically use the collected information to detect the local traffic status and send this information to the authorities. Besides, the information is distributed to the OBUs of other vehicles where it can be used for tasks such as route optimization or other adaptations of driving behavior.

VANET-based traffic information systems have received a lot of attention over recent years (Hartenstein and Laberteaux, 2009; Katsaros et al., 2011). Therefore, a number of applications have been analyzed in the literature, e.g., *enhanced route guidance and navigation*, *congested road notification*, *green light optimal speed advisory*, and *parking availability notification*.

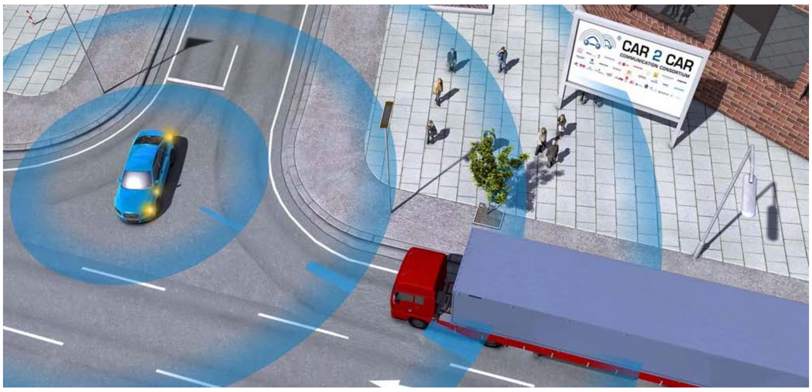
In terms of data dissemination, most of these applications require V2V and V2I communications to exchange messages periodically: usually, every second data is forwarded (data exchange frequency of 1 Hz) (Al-Sultan et al., 2014). Even though, there are applications that include also V2B in order to transport the information to the authorities and governmental institutions (Gerla and Kleinrock, 2011).

The VANET traffic efficiency applications offer a set of important powerful benefits over the classical systems. The most salient advantages are the following ones:

- The information used by the system is collected along the entire road network (if the vehicles travel on the road), not just at certain limited points.

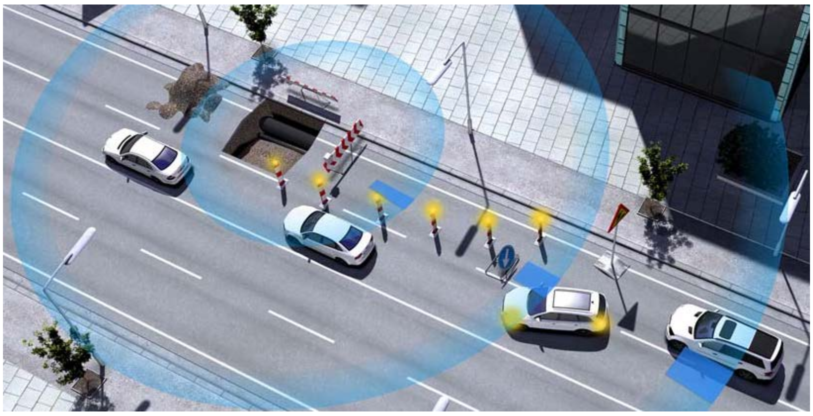
- OBUs always have updated information about the status of the traffic (region-wide, city-wide, etc.) and they are warned by their neighbor nodes of any incidents in real time.

- The driver may be personally warned just if it is necessary according to its profile and the travel destination, reducing possible unnecessary distractions by useless notifications.



a) Left turn assistant VANET application.

b) Approaching emergency vehicle warning VANET application.



c) Work zone warning VANET application.

FIGURE 2.4: Examples of VANET safety applications (C2C-CC, 2015).

2.3.3 VANET Comfort and Entertainment Applications

Finally, vehicular networks can be used to deploy applications and services to improve the comfort and the entertainment of the drivers and the passengers. These applications consist of quite a diverse set of services (Al-Sultan et al., 2014): *Internet access* for the passengers of the vehicle; *automatic payment* applications, such as automatic tooling or parking payment; specific *geo-located services*, such as point-of-interest notification or geo-located service announcements (e.g., deals of the day of a close shop, cinema showtime, etc.); *entertainment and interactive multimedia services*, such as downloading contents (e.g., music, radio programs, movies, etc.) or playing online games; to name a few. The CARLINK project proposed multiplayer games for the car passengers of different vehicles via V2V in order to ease the uncomfortable experience of being stuck in traffic jams by means of entertaining the road users (excluding the drivers) (CARLINK, 2006).

Due to the diversity of applications in this category, a requirement analysis must be done on a case-by-case basis. It is noticeable that the fault tolerance and communication delays requirements of these applications are less strict than the ones of safety and traffic efficiency applications. Moreover, the network load generated by comfort and entertainment applications should not limit the proper operation of the other ones.

2.3.4 Summary of the Main QoS Requirements

Table 2.1 summarizes the main QoS requirements of the three different application classes presented above. These requirements are evaluated in terms of throughput, reliability or delivery ration (in terms of successful and in-time message transmission), and latency (in terms of total transmission delays).

TABLE 2.1: Overview of the main QoS requirements of the studied VANET application classes.

Application class	Main requirements	Connectivity type
Safety	High reliability, low latency (between 10 ms to 1 s)	V2V and V2I
Traffic efficiency	Medium-to-high reliability	V2V, V2I, and V2B
Comfort and infotaintment	High throughput, medium-to-low latency	V2V, V2I, and V2B

2.4 Unique VANET Features

VANETs have typically been seen as a special case of MANETs (particularly at their beginnings) in which the nodes are basically vehicles (OBUs) and road-side elements (RSU). However they behave fundamentally different principally due to the high mobility of the VANET nodes compared to MANETs and the strict requirements of the ITS applications (Campolo et al., 2015). Specifically, the major unique characteristics of VANETs are presented here:

- *Highly dynamic topology*: VANET topology is continuously changing due to limited communication range of the VANET nodes (hundreds of meters) and the high relative speed between the cars. The life time of the link between vehicles moving in opposite directions is very short in comparison with the case of nodes moving in the same direction. This makes packets forwarding very hard because the network routing paths are very frequently changing (Dua et al., 2014; Chen et al., 2011). Besides that, the VANET topology of the network may be affected by the response to the drivers after receiving messages. For example, a vehicle could stop following a fellow car because it has received a message about the need to refill up the tank just before arriving a gas station, and therefore, the communication link between the two vehicles is broken.

- *Frequent disconnected network*: VANET nodes are subject to continual loss of network connection because the high speed of movement. The likelihood of disconnection increases as the network density decreases. This issue must be addressed in order to deploy services that require ubiquitous connectivity, a probable solution is to increase the density of installed RSUs (Silva et al., 2015) or to use V2B communications (Hossain et al., 2010).

- *Predictable mobility*: Vehicular networks differ from MANETs in which the nodes may move in a random way, because the movement of vehicles is limited. The vehicular mobility is constrained by the road layout and topology, by the need of interacting correctly with other vehicles, and by the obligation to follow traffic rules. Thus, having this complete information it is possible to predict and to model the mobility of the nodes. Some authors have used this mobility modeling and prediction to design robust VANET communication protocols (Menouar et al., 2007; Lai et al., 2009).

- *Different communication conditions*: VANETs may be deployed in different kinds of environments, such as urban areas or highways. At cities, the existence of buildings, trees, and other obstacles limit the signal propagation of the wireless signal, and therefore, the QoS of the network. However, at highways there are better conditions for wireless communication since they are typically built at open areas (here the problem is the high speed of the vehicles). Additionally, the road traffic depends on the moment of the day, e.g., during peak hours at morning the traffic density at the entrances to industrial areas is heavier than at noon. Thus, VANETs must be designed to provide a competitive performance over different conditions that are principally defined by the environment, where they are deployed, and by traffic density (Chen et al., 2011).

- *Infrastructure access*: VANET architecture includes roadside infrastructure elements installed along the roads, i.e., RSUs. This fixed infrastructure is used to provide specific services via V2I or V2X communications, e.g., sign extension applications, or to extend network connectivity, such as allowing connection to the Internet (Campolo et al., 2015). Classic MANETs do not consider this solution.

- *Hard delay and delivery constraints*: As it has been already discussed in Section 2.3, VANET safety applications, which are necessary in preventing hazardous situations and accidents, have high requirements with respect to real-time functionality and reliability. Information included in the safety messages is only meaningful during few tens of milliseconds. Moreover, loss of messages may increment the uncertainty in the system, and therefore, endanger human life of the road users.

- *Data locality*: For a large number of VANET applications, the data produced by vehicles is usually relevant to a certain geographical region of the road network. In general, these applications rely on a broadcast distribution of data where the destination nodes are typically those located close to the source node (Al-Sultan et al., 2014).

All these characteristics cause the classic solutions utilized in MANETs, such as routing and broadcasting protocols, are not be able to be directly applied to vehicular environments. Thus, the research community is facing the impressive challenge of designing, developing, and deploying new approaches to deal with vehicular communications by using VANETs.

2.5 Research Projects and Consortia

The high benefit of having a platform for vehicular communications to deploy ITS motivate the development of many different projects, in which the main aim is the design, deployment, and analysis of such communication networks. These projects have been conducted by governmental institutions in Europe, United States, and Asia. In addition they have involved many car producing companies like BMW, Ford, Volvo, etc. and other type of industrial partners.

The most known pioneering vehicular communication activities for each geographical area are: *C2C-CC* project in Europe (C2C-CC, 2007), *VSC (Vehicle Safety Communication)* is a project in the USA (VSC, 2005), and *ASV (Advance Safety Vehicle Program)* project in Japan (Mashita, 2003). Figure 2.5 illustrates a representative set of projects (Zeadally et al., 2012; NEO, 2015).

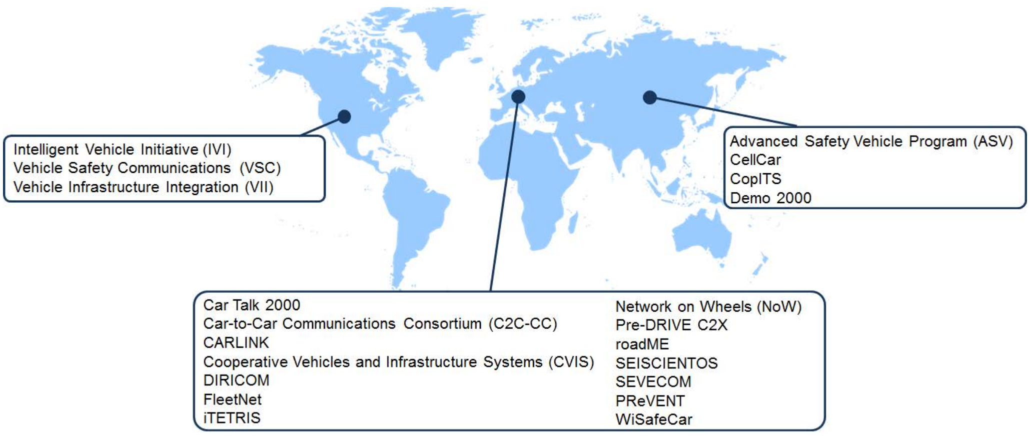


FIGURE 2.5: A representative example of VANET projects and consortia.

2.6 Open Challenges

The projects presented above put much effort into overcoming the challenges that deploying an efficient and reliable vehicular communication platform presents. The main key challenge is that no communication coordinator (central entity) can be assumed. Although some applications involve infrastructure (e.g., traffic signal violation warning or toll collection), most of applications are expected to operate over reliable ad hoc direct communications. This is the essence of the challenges that VANET pose to the research community and industry. Let us introduce now the main research challenges addressed in this domain at world level:

- *Data routing*: Many of VANET applications require to forward data packets from a single source node to a single or multiple destination nodes, e.g., the enhanced route guidance and navigation application. Routing in VANETs has been widely investigated in the past few years (Lee et al., 2010; Chen et al., 2011). Initially, commonly used MANET routing protocols were evaluated for use in VANETs due to the similarities between these two kinds of ad hoc networks. However, some of the specific characteristics of VANETs, such as the highly dynamic topology, the frequent disconnected network or the variability of the network density, make the use of conventional MANET routing protocols inadequate for vehicular environments. Thus, much research effort has been devoted to design more suitable routing strategies for VANETs (Toutouh et al., 2012b; Patil and Dhage, 2013). Figure 2.6 summarizes the a widely used taxonomy in VANET routing (Dua et al., 2014).

- *Beacon broadcasting*: Vehicles in a VANET periodically broadcast short data packets with their location and different kind of kinematic information (geographic location, velocity, etc). These packets are known as *beacons* and they are used as information source for most cooperative driving applications, also known as *cooperative vehicle safety* (CVS) (Sengupta et al., 2007). For this reason, the performance of periodic beacon broadcasting in terms of reliability and latency is a big concern in VANET research, since this exchange of information is one of the keys to improve safety in the road transport. However, this approach is limited because the probability of suffer from network congestion problems increase with the road traffic density which lowers the correct beacon delivery and increases the message delivery times (Fallah et al., 2010). This increment in the communication delays is an important concern in VANETs because beacons have a limited lifetime. The vehicle's status information stored in a beacon is only useful until the next beacon is generated, in order to maintain updated the information received by the nearby vehicles for traffic safety. In the recent literature specific broadcasting methods for vehicular communications are proposed (Chen et al., 2010; Sattari et al., 2012).

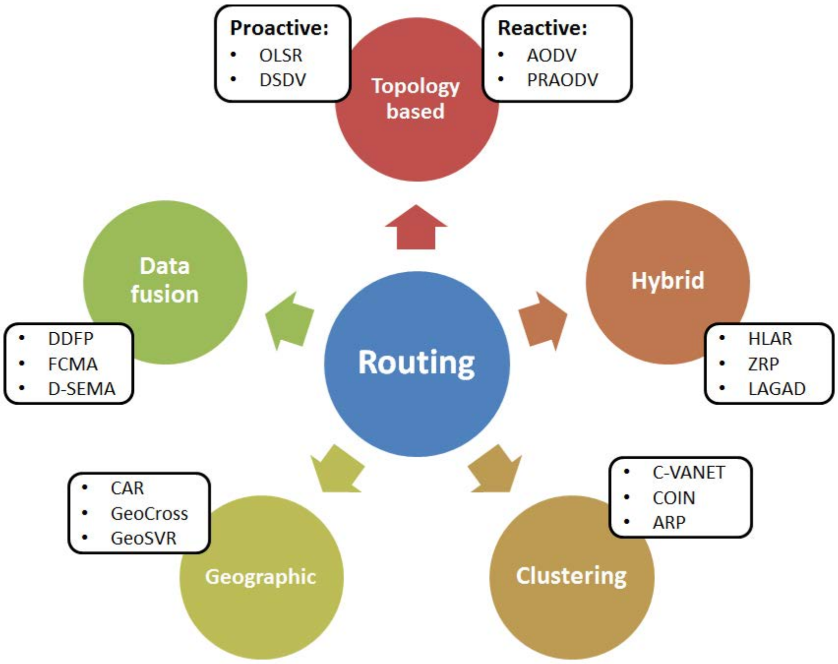


FIGURE 2.6: Some examples of VANET routing algorithms analyzed in Dua et al. (2014).

- *Heterogeneous connectivity*: VANETs originally emerged taking into account just IEEE 802.11 based technologies to exchange data via ad hoc networking, i.e., by using V2V and V2I communications (Hartenstein and Laberteaux, 2009). However, the increasing number of wireless communication technologies and standards have brought immense opportunities and challenges to provide seamless connectivity in vehicular networks. Thus, researchers defined *advanced heterogeneous vehicular networks* (AHVNs), also known as *hybrid vehicular networks* (HVNs), as vehicular networks that use multiple RATs (DSRC, WiMAX, LTE, ...) in a collaborative manner (Hossain et al., 2010). The main challenge in designing AHVNs is to efficiently decide which RAT to select to send a given packet, due to sheer number of use cases and applications with diverse and stringent QoS performance requirements and different communication conditions (Mir et al., 2015). For example, Benslimane et al. (2011) proposed a cluster-based strategy to deploy heterogeneous vehicular networks over Wi-Fi and 3G access technologies.

- *Hardware platform deployment*: Vehicular communications may suffer from very frequent network disconnection during low density road traffic situations, i.e., in sparse VANETs. This can critically limit the network performance, and therefore, it adversely affect the proper operation of the VANET safety applications. The deployment of a fixed infrastructure along the roads composed by RSUs, that act as VANET base stations, may greatly improve the communications by increasing the overall coverage of the network (Reis et al., 2014). However, the deployment of the RSU infrastructure is a hard-to-solve (*NP-hard*) problem (Trullols et al., 2010) because the designers have to decide about the location of the RSUs and the utilized hardware for each RSU, while not incurring to high deploying costs. There are different research lines to deal with this problem, e.g., some authors have model RSU deployment problem as a weighted approach for the traditional *Maximum Coverage Problem* (MCP) (Silva et al., 2015).

- *Security and privacy*: In VANET safety applications, drivers may take life-critical decisions and actions based on the information received from other nodes. Therefore, any malicious entity could cause disruption of traffic, dangerous situations, and accidents by modifying and replying the disseminated messages with fake information. In contrast, users are very conservative about sharing their privacy-related information, e.g., having their driving routes unconditionally accessed by the public. Thus, it is imperative that the VANET system must be able to determine the reliability of the users (drivers) and the messages while still maintaining their privacy and anonymity. In the literature, there are many studies dealing with this concern (Mejri et al., 2014). For example, Molina-Gil et al. (2014) proposed a new data aggregation protocol for vehicular networks, which uses probabilistic verification to detect malicious behavior of users. At the present time, the IEEE 1609.2 standard addresses the issues of securing WAVE messages against the possible malicious attacks and it is positioned for providing secure services for WAVE communications (Lin et al., 2008).

- *Network performance evaluation*: The evaluation of the different implemented approaches (radio technologies, protocols, and network models) to deploy vehicular networks is a major concern in the field of VANETs. Outdoor testbeds can be undertaken to accomplish this task. These experiments may be carried out in real world environments offering close-to-real or real performance (*in vitro versus in silico*), as well as revealing behavioral issues (Santa et al., 2008). However, there is a lack of scientific articles that use outdoor experiments in the field of VANETs. The main reasons for this may be the unavailability and the cost of resources (high number of vehicles and road equipment). Thus, simulation and emulation tools are widely used to overcome the limitations of outdoor experiments in vehicular networks (Martinez et al., 2011). Authors have used three different types of strategies to simulate VANETs (Alba et al., 2008): utilizing a well-known network simulator, such as ns-2/ns-3 (NS2, 2015; Riley and Henderson, 2010; Issariyakul and Hossain, 2008) or OPNET (Sethi and Hnatyshin, 2012), that allows users to define the movement of the nodes by generating realistic VANET traces with transportation simulators, such as (Härri et al., 2011), SUMO (Krajzewicz et al., 2012) or VISSIM (Lownes and Machemehl, 2006); coupling vehicular traffic and wireless network in a single simulator, e.g., GrooveNet (Mangharam et al., 2006); and the most promising approach, synchronizing the existing and validated traffic and network simulators by using some specific bridge software, such as iTetris (Rondinone et al., 2013) and Veins (Sommer et al., 2011). Even if the simulators have achieved a high degree of realism and they obtain quite accurate results, the real world simplifications that they apply could limit their reliability.

As it can be inferred from the previous list, this domain still comprises a heterogeneous set of open questions that have to be resolved in order to provide a body of knowledge to advance in vehicular communications. This PhD thesis aims at analyzing the application of **Natural Computing** (NC) in addressing a subset of these open challenges (see Section 3.3). Thus, the following chapters present the NC techniques utilized and how they are applied in dealing with data **routing**, beacon **broadcasting**, and designing the **roadside infrastructure** challenges. In addition, real world VANET/ITS **applications and demonstrations** are described. The idea is to make a journey from fundamental to more practical aspects related with vehicular communications, that have been selected because we thought that they are in the center of the worries in the whole domain.

Chapter 3

Natural Computing and Optimization Challenges in VANETs

NATURAL computing (NC) encompasses different classes of methods inspired in nature. Among others, it includes a novel methodology of complex problem solving techniques (e.g., evolutionary algorithms or swarm intelligence). In this thesis, we focus on the use of these techniques to address different VANETs questions. This chapter summarizes the basics about optimization problem solving, presents the NC algorithms utilized here, and introduces the reader to the different VANETs problems analyzed, including the most salient related works. Finally, it illustrates the process performed to evaluate NC in solving VANET optimization problems.

3.1 Introduction

Natural phenomena, e.g., processes, organisms, etc., have long inspired and driven people to mimic, design, and develop systems and products. ***Natural Computing*** (NC) is the computational process of developing artificial (computational) systems by extracting ideas from nature or by using natural media (such as molecules). NC are generally divided in three main branches (De Castro, 2006): *computing inspired by nature*, *the simulation and emulation of nature by means of computing*, and *computing with natural materials*.

In this thesis we focus on the first branch, computing inspired by nature, which globally takes the inspiration from nature (natural patterns, behaviors, and organisms) to design algorithms for the solution of complex problems. The computational techniques developed under this umbrella can also be termed as *bio-inspired computing* (Mange and Tomassini, 1998) or *computing with biological metaphors* (Paton, 1994).

The landmark in the bio-inspired computing was the paper by McCulloch and Pitts (1943), which laid the foundations of *artificial neural networks* by introducing the first mathematical model of a neuron. Afterwards, another computation approaches emerged inspired by nature. These other techniques are grouped in three main types: **1)** *evolutionary computing*, which uses the ideas from evolutionary biology to design *evolutionary algorithms* (EAs); **2)** *swarm intelligence*, in which a set of simple agents mimic the behavior of social organisms in a given algorithm; and **3)** *artificial immune systems* (AIS), which extract ideas from the models followed by immune systems to develop computational tools, cover one of the last groups to appear. In addition, several emerging bio-inspired algorithms can be found in the literature that specifically do not belong to any of the previous types, e.g., *simulated annealing* (SA).

This thesis aims at addressing a set of the main challenges that limit the VANETs deployment (see Section 2.6). These challenges may be tackled as *hard to solve* optimization problems, which can be solved by utilizing NC. Specifically, in our research we have focus on the use of EAs, swarm intelligence, and SA in solving VANETs optimization problems. Now let us introduce some essential concepts about optimization problems.

3.1.1 Overview of Optimization Problems

There exist a large number of real-life problems that are complex and difficult to solve. *Exact algorithms* are not appropriate or require large amount of resources (e.g. memory or computational cost) for using them. Therefore, approximate algorithms are needed. Among approximate algorithms, one can find two types: *heuristics* and *metaheuristics*. Heuristics can in turn be divided between *constructive heuristics* and *local search* methods. We focus this chapter on metaheuristics. Figure 3.1 shows a simple classification of optimization methods used throughout the history of computer science.

Metaheuristics are approximate algorithms that emerged as efficient (stochastic) optimization tools that are able to provide good solutions for complex optimization problems (Glover and Kochenberger, 2003; Blum and Roli, 2003). In general, metaheuristics make no assumptions about the problem to solve, as generic tools they only use appropriate solution representations, quality or fitness functions, and (specific) operators to guide the solution towards better solutions (Osman and Kelly, 2012). Many metaheuristics are inspired by nature, such as EAs, swarm intelligence, and SA (Figure 3.1) (Yang, 2010). Before detailing these NC methods we will introduce some essential concepts about optimization.

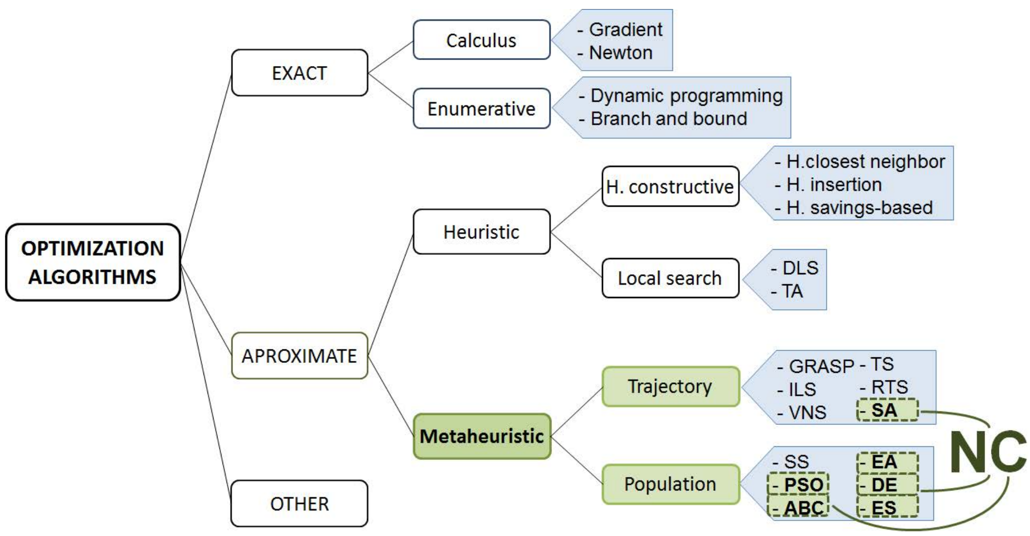


FIGURE 3.1: General classification of the optimization techniques.

We shall begin with a formal definition of *optimization*. Assuming, without loss of generality, a *minimization case*, the definition of an optimization problem is as follows:

Definition 1. Optimization problem An optimization problem is defined as a pair (S, f) , where $S \neq \emptyset$ is called the *solution space* (or *search space*) of the problem, while f is a quality criterion named *objective function* or *fitness function*, defined as:

$$f : S \longrightarrow \mathbb{R} \tag{3.1}$$

In this case where a single criterion f is optimized, known as *mono-objective* or *single-objective optimization*, the objective is to find a *global optimum* element $s^* \in S$ such that:

$$f(s^*) \leq f(s) \;\; \forall s \in S \tag{3.2}$$

Depending on the domain where S belongs, we can speak of *binary* ($S \subseteq \mathbb{B}^*$) , *integer* ($S \subseteq \mathbb{N}^*$), *continuous* ($S \subseteq \mathbb{R}^*$) or *heterogeneous* ($S \subseteq (\mathbb{B} \cup \mathbb{N} \cup \mathbb{R})^*$) optimization problems.

Note that assuming either maximization or minimization does not restrict the generality of the results, since an equivalence can be made between the two cases in the following manner (Goldberg, 1989):

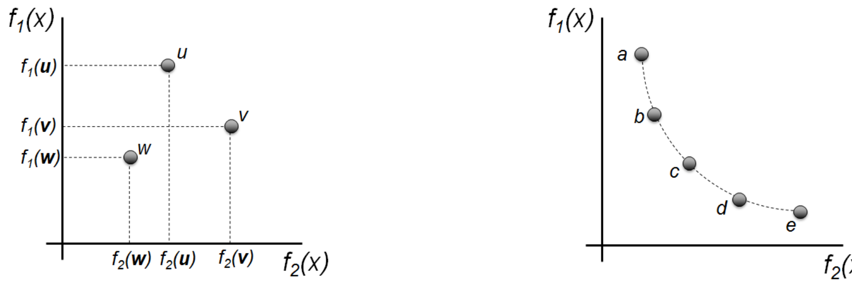
$$max\{f(s) \mid s \in S\} \equiv min\{-f(s) \mid s \in S\} \tag{3.3}$$

This definition is utilized when the optimization problem focus on a single objective (mono-objective). Nevertheless, many the real-world problems deal with different objectives that are usually in conflict with each other (e.g. maximizing the coverage of the VANET infrastructure elements but while minimizing the cost of the installed infrastructure). This other types of problems are known as *multi-objective optimization* (Deb, 2001). The main difference between mono-objective and multi-objective optimization is that for the second ones there is not a single optimal solution that satisfies all the objectives but a set.

A general *multi-objective optimization problem* (MOP) is to find vectors $\vec{s^*} = [s_1^*, s_2^*, ..., s_n^*]$ that are optimizing the vector of functions $\vec{f}(\vec{s}) = [f_1(\vec{s}), f_2(\vec{s}), ..., f_k(\vec{s})]$. Each $f_i(\vec{s})$ is a mono-objective optimization problem, and it is considered one of the objectives to optimize in our MOP. The different objectives must be in conflict with the others, meaning that an increase in the quality of one of them will lead to a decrease in the values of (some of) the others. If the objectives were not in conflict, then we could reformulate the problem as a mono-objective one. More formally multi-objective minimization problem is defined as:

Definition 2. Multi-objective minimization Find a vector $\vec{s^*} = [s_1^*, s_2^*, ..., s_n^*]$ which satisfies the m inequality constraints $g_i(\vec{s}) \geq 0$, $i = 1, 2, ..., m$, the p equality constraints $h_j(\vec{s}) = 0$, $j = 1, 2, ..., p$, and minimizes the vector function $\vec{f}(\vec{s}) = [f_1(\vec{s}), f_2(\vec{s}), ..., f_k(\vec{s})]^T$, where $\vec{s} = [s_1, s_2, ..., s_n]^T$ is the vector of decision variables.

In MOP, to decide whether a given solution is better than other is utilized the concept of *dominance*. A solution w dominates a solution v if w is strictly better than v in at least one objective and better or equal to v in the rest of objectives (see Figure 3.2.a). A set of solutions are said to be *non-dominated* if none dominates the others. Therefore, the goal of multi-objective optimization is to find the optimal set of non-dominated solutions to the problem, which is named *Pareto optimal set* (see Figure 3.2.b). The projection of the Pareto optimal set in the objectives domain is called the *Pareto optimal front* (Coello et al., 2007).



a) The w solution dominates u and v. b) Non-dominated solutions.

FIGURE 3.2: Dominance in multi-objective optimization.

After these definitions let us presenting the NC algorithms applied in this thesis.

3.2 NC Applied in VANETs

The bio-inspired algorithms applied in this research are categorized into EAs, swarm intelligence, and SA. **Evolutionary algorithms** are based on the Darwinian theory of evolution (Fogel et al., 1966; Holland, 1975). Darwin proposed that a population of individuals capable of reproducing and subjected to (genetic) variation followed by selection result in new populations of individuals increasingly more fit to their environment (De Castro, 2006). These simple natural processes when applied in computation result in a number of different algorithms, such as: *genetic algorithms* (GA) (Goldberg, 1989), *evolution strategies* (ES) (Beyer and Schwefel, 2002), *evolutionary programming* (EP) (Fogel, 1999), and *genetic programming* (GP) (Koza, 1992).

Swarm intelligence was introduced to refer to cellular robotic (multi-agent) systems in which a collection of simple agents in an environment interact based on local rules (Bonabeau et al., 1999). Currently, this term is used to describe the design of algorithms or problem-solving devices inspired by the collective behavior of social organisms. Some swarm intelligence algorithms used to address hard to solve optimization problems are: *particle swarm optimization* (PSO) (Eberhart and Kennedy, 1995), *ant colony optimization* (ACO) (Dorigo et al., 1996), and *artificial bee colony algorithm* (ABC) (Karaboga and Basturk, 2007).

Finally, **simulated annealing** (SA) is one of the oldest bio-inspired algorithms and it is based on the annealing process of metal and crystal (Kirkpatrick et al., 1983). It is considered as the first method with an explicit strategy for escaping local optima.

These natural inspired algorithms can be classified as *trajectory based* algorithms and *population based* or *swarm based* ones (see Figure 3.1). Those of the first type handle a single element of the search space at a time (one solution), e.g. SA; while those of the latter work on a set of elements (named population or swarm), the EAs and swarm intelligence algorithms utilized in this thesis belong this group. We will now describe the NC algorithms applied in our research.

3.2.1 Evolutionary Algorithms

Four different EAs have been utilized in this thesis: three mono-objective algorithms, i.e., GA, ES, and *differential evolution* (DE), and one multi-objective EA (MOEA), the *non-dominated sorting genetic algorithm-II* (NSGA-II). They are detailed in the following subsections.

Genetic Algorithm (GA)

Genetic Algorithm (Goldberg, 1989) is the most popular EA. It iterates a process in which a set of solutions (*parents*) are selected from the whole population with a given *selection* criterion, they are then *recombined* (*crossover operation*), the obtained *offsprings* are *mutated*, and finally they are evaluated and inserted back into the population following a given criterion (*fitness function*). The mutation process is carried out by randomly (uniformly) selecting one of the elements in the solution, and assigning (randomly) a new value in a specific range. In this research work, we use a polynomial crossover defined for continuous variables and two point crossover (Goldberg, 1989) as the recombination operator. In addition, specific mutation operators have been designed for the problems addressed here. Algorithm 1 summarizes the operations of a canonical GA.

There are two main versions of GA: *steady state* GA (ssGA) and *generational* GA (genGA). The difference between the ssGA and the genGA is the way in which the population is updated with the new individuals generated during the evolution. In the first one, new individuals are directly inserted into the current population. In the case of the genGA, a new auxiliary population is built with the obtained offsprings and then, once this auxiliary population is full, it completely replaces the current population. Thus, in ssGAs the population is asynchronously being updated with the newly generated individuals, while in the case of genGAs all the new individuals are updated at the same time, in a synchronous way.

Algorithm 1 Pseudocode of GA

$g \leftarrow 0$
$P_g \leftarrow$ **initializePopulation()** // P = population
while not stopCondition() do
  $parents \leftarrow$**selection**(P) // Select parents
  $offspring \leftarrow$**recombination**$(parents)$ // Generate offspring by recombination
  $offspring \leftarrow$**mutation**$(offspring)$ // Generate offspring by mutation
  evaluate$(offspring)$
  $P_{g+1} \leftarrow$ select$(offspring)$ // New population generation
  $g \leftarrow g + 1$
end while

Evolutionary Strategy (ES)

Evolutionary Strategy (Beyer and Schwefel, 2002) is an EA also based on the ideas of adaptation and evolution. As common with EAs, the mutation and selection operators are applied to the solutions (individuals) through a given number of generations. The selection in evolutionary strategies is *deterministic* and only based on the *fitness rankings*. In ES, we use the same mutation operator than the GA.

Algorithm 2 Pseudocode of ES

1: $g \leftarrow 0$
2: $parent_0 \leftarrow$ **initializeParent()**
3: **while not stopCondition() do**
4:   $offspring_g \leftarrow$ **mutate**$(parent_g)$
5:   **evaluate**$(offspring_g)$
6:   **if** $f(offspring_g)$ **is better than** $f(parent_g)$ **then**
7:     $parent_g \leftarrow offspring_g$
8:   **end if**
9:   $g \leftarrow g + 1$
10: **end while**

The canonical ES (Algorithm 2) operates on a population of size two: the current individual ($parent_g$) and the result of its mutation ($offspring_g$). After the parent initialization (Line 2), ES starts the evolutionary process by generating a mutated offspring (Line 4) which is evaluated (Line 5). Only if the offspring has a better fitness than the parent, it becomes the parent of the next generation (lines 6-8). Otherwise the offspring is ignored. This is version of ES is called $(1 + 1)-$ES. More generally, in (1 + λ)-ES, a population with more than one offsprings (λ) can be generated for being compared with the same parent. In a (1, λ)-ES the best offspring becomes the parent of the next generation while the current parent is always ignored. The most generalized version, $(\mu+/, \lambda)$-ES, often uses a population of parents (μ) and also recombination as an additional operator.

Differential Evolution (DE)

Differential Evolution (Price et al., 2005) is also a stochastic population based algorithm designed to solve optimization problems in continuous domains. The main difference in the evolutionary model with the other EAs is that the fittest of an offspring competes one-to-one with that of corresponding parent.

The population consists of a set of individuals (vectors) which evolve simultaneously through the search space of the problem. The task of generating new individuals is performed by differential operators such as the *differential mutation* and *differential crossover*. A mutant individual w^i_{g+1} is generated by the following Equation 3.4:

$$w^i_{g+1} \leftarrow v^{r1}_g + \mu \cdot (v^{r2}_g - v^{r3}_g) \tag{3.4}$$

where $r1, r2, r3 \in \{1, 2, \ldots, i-1, i+1, \ldots, N\}$ are random integers mutually different, and also different from the index i. The mutation constant $\mu > 0$ stands for the amplification of the difference between the individuals v^{r2}_g and v^{r3}_g, and it avoids the stagnation of the search process. In order to increase even more the diversity in the population, each mutated individual undergoes a crossover operation with the *target individual* v^i_g, by means of which a *trial individual* u^i_{g+1} is generated. A randomly chosen vector component is taken from the mutant individual to prevent that the trial individual replicates the target individual.

$$u^i_{g+1}(j) \leftarrow \begin{cases} w^i_{g+1}(j) & \text{if } r(j) \leq C \text{ or } j = j_r, \\ v^i_g(j) & \text{otherwise.} \end{cases} \tag{3.5}$$

As shown in Equation 3.5, for each component j of the trial individual u^i_{g+1}, the crossover operator chooses both, a random integer value j_r and a random real number $r(j) \in (0, 1)$, uniformly distributed. Then, the crossover probability C and $r(j)$ are compared just like j and j_r. If r is less than or equal than C or j is equal to j_r, then we select the j^{th} element of the mutant individual to be allocated in the j^{th} element of the trial individual u^i_{g+1}. Otherwise, the j^{th} element of the target individual v^i_g becomes the j^{th} element of the trial individual. Finally, a selection operator decides the acceptance of the trial individual for the next generation if and only if it yields a reduction (assuming minimization) in the value of the fitness function f, as shown in Equation 3.6:

$$v^i_{g+1} \leftarrow \begin{cases} u^i_{g+1} & \text{if } f(u^i_{g+1}) \leq f(v^i_g), \\ v^i_g(j) & \text{otherwise.} \end{cases} \tag{3.6}$$

Algorithm 3 Pseudocode of DE

1: $g \leftarrow 0$
2: $P_0 \leftarrow$ **initializePopulation()** // P = Population
3: **while not stopCondition() do**
4:     **for each individual** v^i_g **in** P **do**
5:         **chooseMutuallyDifferent**(r_1, r_2, r_3)
6:         $w^i_{g+1} \leftarrow$ **differentialMutation**($v^{r1}_g, v^{r2}_g, v^{r3}_g, \mu$) // Equation 3.4
7:         $u^i_{g+1} \leftarrow$ **differentialCrossover**(v^i_g, w^i_{g+1}, C) // Equation 3.5
8:         **evaluate**(u^i_{g+1})
9:         $v^i_{g+1} \leftarrow$ **selection**(v^i_g, u^i_{g+1}) // Equation 3.6
10:     **end for**
11:     $g \leftarrow g + 1$
12: **end while**

Algorithm 3 shows the pseudocode of DE. After initializing the population, the individuals evolve while stop condition is not reached. Each individual is then mutated (Line 6) and recombined (Line 7). The new individual is selected (or not) following the operation of Equation 3.6 (lines 8 and 9).

Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

NSGA-II is a *multi-objective evolutionary algorithm* (MOEA) version of the GA presented in Deb et al. (2002). It emerged to solve the main drawbacks of its prior version, NSGA, that are: the computational complexity, the lack of elitism, and the need of choosing the optimal parameter value for sharing parameter σ_{share}. Nowadays, NSGA-II is one of the reference algorithms to solve multi-objective problems. Therefore, we have selected it as a baseline for the research work carried out in this thesis.

Its pseudocode is presented in Algorithm 4. NSGA-II makes use of a *population* (P_g) of solutions. In each generation, it creates new individuals (*offspring*) after applying genetic operators to P_g (recombination and mutation) to create a new population Q (lines 6 to 8). Then, both the current (P_g) and the new population (Q) are merged; the resulting population, R, is ordered according to a ranking procedure and a density estimator known as *crowding distance* (Line 13) (Deb et al., 2002). Finally, the population P_g is updated with the best individuals in R (Line 14). These steps are repeated until the termination condition is fulfilled.

Algorithm 4 Pseudocode of NSGA-II.

1: $g \leftarrow 0$
2: $P_0 \leftarrow$ **initializePopulation()** // P_g = population
3: **while not stopCondition() do**
4: $\quad Q \leftarrow \emptyset$ // Q = auxiliary population
5: $\quad$ **for** i $\leftarrow$ 1 **to** (P_g.popSize / 2) **do**
6: $\quad\quad parents \leftarrow$ **selection**(P_g)
7: $\quad\quad offspring \leftarrow$ **recombination**($parents$)
8: $\quad\quad offspring \leftarrow$ **mutation**($offspring$)
9: $\quad\quad$ **evaluate_solution**($offspring$)
10: $\quad\quad$ **insert**($offspring$,Q)
11: $\quad$ **end for**
12: $\quad R \leftarrow P_g \cup Q$ // R = auxiliary resulting population
13: $\quad$ **rankingCrowding**(R)
14: $\quad P_{g+1} \leftarrow$ **selectBestIndividuals**(R) // Applying elitism
15: $\quad g \leftarrow g + 1$
16: **end while**

3.2.2 Swarm Intelligence

Regarding swarm intelligence algorithms, in this PhD thesis we have applied two different methods: PSO to tackle mono-objective problems and *speed-constrained multi-objective particle swarm optimization* (SMPSO) to address MOPs (Nebro et al., 2009). Let us to describe these two algorithms.

Particle Swarm Optimization (PSO)

Particle swarm optimization (Eberhart and Kennedy, 1995) is a NC method inspired in the social behavior of bird flocking or fish schooling. It was initially designed for continuous optimization problems. Each potential solution to the problem is called *particle* and the set of particles is called a *swarm* (hence the name of the algorithm). In this algorithm, each particle position p_g^i is updated each generation g by means of the Equation 3.7:

$$p_{g+1}^i \leftarrow p_g^i + v_{g+1}^i, \tag{3.7}$$

where the term v_{g+1}^i is the velocity of the particle, given by the next expression:

$$v_{g+1}^{i} \leftarrow w \cdot v_{g}^{i} + \varphi_1 \cdot U^{+} \cdot (bp_{g}^{i} - p_{g}^{i}) + \varphi_2 \cdot U^{+} \cdot (b_g - p_{g}^{i}) \tag{3.8}$$

In Equation 3.8, bp_g^i is the best solution that the particle i has stored so far, b_g is the best particle (also known as the *leader*) that the entire swarm has ever created, and w is the *inertia weight* of the particle which controls the trade-off between exploitation and exploration. Finally, φ_1 and φ_2 are specific parameters which control the relative effect of the personal and global best particles (typically, φ_1=φ_2=2). U^+ is a uniform random value $\in (0, 1)$.

Algorithm 5 Pseudocode of PSO

1: $g \leftarrow 0$
2: **initializeSwarm**(S)
3: $b_0 \leftarrow$**locateLeader**(S)
4: **while not stopCondition() do**
5:     **for each particle** x_g^i **in** S **do**
6:         **updateVelocity**(v_g^i) // Equation 3.8
7:         **updatePosition**(p_g^i) // Equation 3.7
8:         **evaluate**(p_g^i)
9:         **update**(bp_g^i)
10:     **end for**
11:     $b_{g+1} \leftarrow$**updateLeader**(S)
12:     $g \leftarrow g + 1$
13: **end while**

Algorithm 5 describes the pseudocode of general PSO. It starts by initializing the swarm S (Line 2), which includes the positions, velocities, and corresponding bp^i of each particle. Then, the leader b_0 is also initialized (Line 3). During iteration and while the stop condition is not reached, each particle *flies* through the search space updating its velocity and position (lines 6 and 7), it is then evaluated (Line 8), and its bp^i is also calculated (Line 9). At the end of each iteration, the leader b_g is updated (Line 11).

Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO)

MOAs inspired on PSO quickly became very popular because of their competitive performance in solving continuous problems. However, the early approaches were unable to solve multi-frontal problems satisfactorily due to the *swarm explosion* problem, whereby velocity of the particles becomes too high resulting in erratic particle movements. SMPSO applies a velocity constriction method to mitigate this problem (Nebro et al., 2009).

The SMPSO pseudocode is presented in Algorithm 6. As in PSO, the operations are performed over particles (p^i) of a swarm (S). The non-dominated solutions are stored in the *leaders archive* (L_g), which is initialized in Line 3 and updated in Line 11 by using *crowding distance*. A loop over all particles of the swarm defines the main operations to be carried out. The *turbulence operator* is a mutation applied after updating the particles speed and position (lines 6-8) (Nebro et al., 2009). Then, the particles are evaluated (Line 9). Finally, the memory of each particle is updated (Line 12).

Algorithm 6 Pseudocode of SMPSO.

```
g ← 0
initializeSwarm(S)   // S = swarm
initializeLeadersArchive(L_0) // L = leaders
while not stopCondition() do
    for each particle p^i in S do
        computeSpeed(p^i)
        updatePosition(p^i)
        mutation(p^i)
        evaluateSolution(p^i)
    end for
    L_{g+1} ←updateLeadersArchive()
    updateParticlesMemory()
    g ← g + 1
end while
```

3.2.3 Simulated Annealing

SA was first proposed by Kirkpatrick et al. (1983). It is inspired in the industrial processes of annealing, and basically lies in a local search method with a mechanism that eventually promotes solutions of worse quality than the current ones (*uphill moves*) in order to escape from local minima. It is a fairly commonly used algorithm that provides good results and constitutes an interesting method to compare to other optimizing methods because of its simplicity. The pseudocode for this algorithm is shown in Algorithm 7.

Algorithm 7 Pseudocode of SA

```
1:  initialize(T,s)
2:  evaluate(s)
3:  while not stopCondition() do
4:      while not coolingCondition(g) do
5:          s' ← chooseNeighbor(s) // Generate new solution
6:          evaluate(s')
7:          if accept(s,s',T) then // Acceptance criterion
8:              s ← s'
9:          end if
10:     end while
11:     coolDown(T) // E.g. T = T × 0.999
12: end while
```

The whole process starts by generating an initial solution s and starting the *temperature* parameter (T). The algorithm works keeping a single tentative solution s at any time, and therefore, it is a trajectory based method. In every iteration, a new solution s' is generated from the previous one (Line 5), and either replaces it or not depending on an acceptance criterion (lines 7-9). The acceptance criterion works depending on the fitness values ($f(s)$ and $f(s')$) and temperature T. The new solution, s', replaces s if s' has better quality, otherwise s' replaces s according to the probability $prob$ (see Equation 3.9). This probability depends on the difference between their quality ($f(s') - f(s)$) values and T (Line 9). This acceptance criterion provides the way of escaping from local optima.

$$prob(T, s, s') = \frac{2}{1 + e^{\frac{f(s')-f(s)}{T}}} \tag{3.9}$$

As iterations go on, the value of the temperature (T) is reduced following a cooling schedule (Line 11), thus biasing SA towards accepting only better solutions. In this thesis, we employ the geometric rule $T(n+1) = \alpha \cdot T(n)$, where $0 < \alpha < 1$, and the cooling is performed every k iterations (k is the *Markov chain length*).

3.2.4 Parallel NC algorithms

As it will be seen in Chapter 4, one of the main issue when solving off-line VANET optimization problems is that solution evaluations (fitness computations) require VANETs simulations, which are computationally expensive. This limits the effectiveness of NC in finding competitive solutions in reasonable execution times. Therefore, we have analyzed the use of parallel NC algorithms in order to address this issue.

The relatively inexpensive cost of parallel computing platforms leads researchers to utilize parallel and distributed processing to deal with the time/resources limitation problems. Parallel implementations of bio-inspired algorithms became popular in the last decade as an effort to improve their efficiency. By splitting the population/swarm or the fitness function evaluation into several processing elements, parallel NC allow reaching high quality results in a reasonable execution time even for hard-to-solve optimization problems (Alba, 2005). The parallel NC algorithms proposed here are categorized within the *master-slave* model according the classification by Alba and Tomassini (2002).

The master-slave model follows a classic functional decomposition of the EA, i.e., *parallel EA* (pEA) or swarm intelligence algorithms, where different stages of the iterative process are performed in several computing resources. The evaluation of the fitness function is the main candidate to perform in parallel, since it usually requires larger computing time than the application of the variation operators. Thus, as it is illustrated in Figure 3.3, our master-slave parallel NC algorithms are organized in a hierarchic structure: a master process performs the main operators of the search process and controls a group of slave processes that evaluate the fitness function.

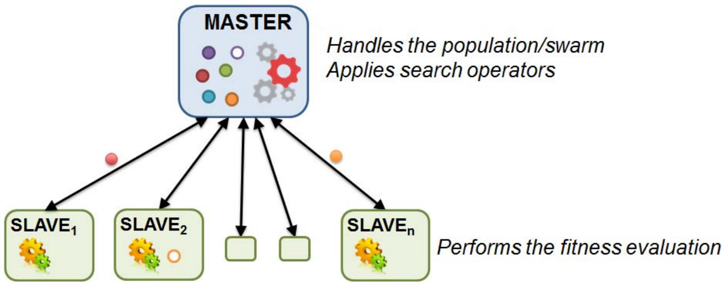


FIGURE 3.3: Master-slave model for parallel NC utilized in this thesis.

3.3 Challenges Tackled in VANETs Using NC

Bio-inspired computation has been applied in many different areas (Chiong, 2009), such as smart mobility (Stolfi and Alba, 2013), telecommunications (Yang et al., 2015), civil engineering (Zavala et al., 2014) or medicine (Dybowski et al., 1996). After the analysis of the different challenges arising from VANETs (see Section 2.6), we suggest NC as an efficient tool to address these complex challenges. Specifically, we have analyzed the application of NC in: *data routing*, *beacon broadcasting*, and *hardware platform deployment*.

3.3.1 Natural Computing for VANET Data Routing

As presented in Section 2.4, some of the main characteristics of the VANETs are: **a)** the high dynamism in topology changes, **b)** the frequent disconnections suffered by the nodes, and **c)** the hard QoS (delay and delivery) constraints. These characteristics critically difficult the **data routing** in such kind of networks. Therefore, a number of specific routing approaches have been proposed in the last years (Chen et al., 2011; Dua et al., 2014; Lee et al., 2010; Santa et al., 2009; Ding et al., 2011; Chauhan and Dahiya, 2012; Campolo et al., 2015).

Recently, bio-inspired methods have been utilized to address the VANET routing problem with the aim of ensuring an optimized QoS (S. Bitam, 2014). The strength of this approach is principally related with the potential of NC in terms of scalability, self-organization, and robustness. For this reason it has been already applied in other types of networks (Dressler and Akan, 2010; Villalba et al., 2010; Dorronsoro et al., 2014). There are two main approaches in using NC in VANET routing: *on-line optimization* and *off-line optimization*. **On-line optimization** applies NC directly in the protocol operation, hence resulting on the design of bio-inspired routing algorithms. The main motivation behind this approach stems from the supposed similarity between the VANET communication scenarios and the natural communication of species. However, these proposals are very difficult to use in real world networks because they require intensive computation and/or some extra infrastructure networks elements (Bitam et al., 2015). **Off-line optimization**, which have been analyzed in-depth in Chapter 4, consists in using the power of NC in solving hard search-based optimization problems to find optimized configurations of software routing protocols. The optimized configurations improve the QoS, while preventing high resources consumption in data routing. The main advantage of this approach is that it can be applied to any kind of routing protocol, even to on-line optimized protocols.

Some of the most outstanding examples of bio-inspired routing algorithms (**on-line optimization**) in the literature use the strength of the *pheromone*-based exchange of information utilized in ACO. For instance, the *Mobility-aware Ant colony optimization Routing DYMO* (MAR-DYMO) designed by Correia et al. (2011), in which ACO is applied to the DYMO (*Dynamic MANET On-demand*) protocol (C. Perkins, 2013) to predict the mobility of the vehicles in terms of speed and position, in order to find efficient routing paths (optimized lifetime). It includes pheromone information in the routing table. MAR-DYMO outperforms DYMO, but at the expense of increasing routing overhead. More ACO-based VANET routing algorithms can be found in the current state-of-the-art, such as the *Vehicular routing protocol based on Ant Colony Optimization* (VACO) (Li and Boukhatem, 2013), *HOPNET* (Wang et al., 2009), and the *Mobility Aware Zone based Ant Colony Optimization Routing for VANET* (MAZACORNET)(Rana et al., 2013). Besides, other swarm intelligence techniques have also been applied, such as ABC or PSO. For instance, the *Quality of Service Bee Swarm routing protocol for VANET* (QoSBeeVANET) (S. Bitam, 2011) and the *Hybrid Bee swarm Routing* (HyBR) designed by Bitam et al. (2013).

Genetic algorithms have been also utilized for the same purpose. The *Adaptive Message Routing* (AMR) proposed by Saleet et al. (2009), which utilizes GA for searching routing paths that minimize (end-to-end) communication delays. The main issues of AMR are: **1)** the increment of the complexity (computation time) of finding optimized routing paths, **2)** the application of crossover and mutation operators may rise to longer routes than the actual ones, and **3)** it assumes the existence of a fixed RSU in the center of each intersection. The *Intersection-based Geographical Routing Protocol* (IGRP) presented in Saleet et al. (2011) utilizes GA to optimize the QoS in terms of delay, bandwidth, and error rate for communications from vehicles to Internet through RSUs. As AMR, one of the main drawbacks of this routing algorithm is the requirement of a preinstalled infrastructure of RSUs.

Regarding to **off-line optimization**, a similar idea of using NC has been applied in optimizing in MANETs (Dorronsoro et al., 2014; Cheng and Yang, 2010; Ruiz, 2011). If we focus exclusively on the use of NC to optimize VANET routing, just a few studies can be found in the literature. Most of them evaluate the QoS improvement in terms of reliability (packet delivery), communication delay, and routing overhead. Mono-objective optimization has been applied to find optimized configuration parameters of OLSR by using sequential (one-thread) PSO (Zukarnain et al., 2014). Similar analysis was carried out by García-Nieto and Alba (2010) to optimize AODV. In this preliminary study, the authors evaluated the use of a set of NC algorithms in addressing VANET routing optimization. In addition, Said and Nakamura (2014) proposed an asynchronous pEA based on hybridization of GA to deal with the optimization of AODV as well. The reduced number of studies in the literature (S. Bitam, 2014) and the quite possible benefits of apply NC in VANET routing off-line optimization motivate us to perform the in depth study that can be found in Chapter 4.

3.3.2 Natural Computing for VANET Beacon Broadcasting

Most of safety related applications presented in Section 2.3 rely on an efficient beacon broadcasting (Hartenstein and Laberteaux, 2009; Campolo et al., 2015). However, VANETs suffer from network congestion (e.g., *broadcasting storm problem*) when the number of nodes increases (Wisitpongphan et al., 2007). Thus, congestion control is a critical research issue that aims at providing reliable environments in modern network communications (Lochert et al., 2007). The reliability of safety applications, that could be the difference between of saving lives or not, is highly dependent on the packet loss and the communication delays. Congestion, which occurs when the network load exceeds the capacity of the network links, generally leads to an increase of both, the packet loss and the communication delays.

Several strategies have been proposed to avoid congestion problems in vehicular communications, while keeping the communication capabilities of the nodes over a given QoS threshold. Most of them can be included in the following basic schemes (Sattari et al., 2012): **i)** adapting the transmission range of transmission channels (Artimy et al., 2005; Mittag et al., 2008; Torrent-Moreno et al., 2009), **ii)** adjusting the data rate generation of applications and services (Xu and Barth, 2004; Schmidt et al., 2010; Rezaei et al., 2007), **iii)** hybrid methods by combining the two previous schemes (Djahel and Ghamri-Doudane, 2012; Tielert et al., 2013), and **iv)** scheduling data packets in various channels based on their priorities, resources, etc (Olariu and Weigle, 2009; Park et al., 2014; Vinel, 2012). These strategies can be studied as (off-line and on-line) optimization problems as well.

Taking into account just **on-line broadcasting optimization** methods, there are some studies that have introduced the use of (non bio-inspired) metaheuristics to define hybrid congestion control methods. In this sense, they utilized *Tabu Search* (TS) (Glover, 1989) to find optimized parameterizations (values for transmission range and transmission rate), after congestion situation is detected. Taherkhani and Pierre (2012) proposed the use of mono-objective TS to minimize just the communication delay. Later, the same authors utilized a multi-objective TS to minimize the communication delay and the jitter (Taherkhani and Pierre, 2015). The main issue of these approaches is the relatively high computation complexity (run times) that has to be considered as an additional time of the final communication delays.

Most work related with **off-line broadcasting optimization** has been applied in urban MANETs (Dorronsoro et al., 2014), which can be seen the root of the current VANETs. One of the most outstanding contributions in the early literature in this domain is Alba et al. (2005), who proposed the optimization of the *Delayed Flooding Cumulative Neighborhood* broadcasting protocol (Hogie et al., 2004). This study applied a specialized cellular multi-objective GA to optimize the coverage, the network use, and the broadcasting time. Later, other studies analyzed different multi-objective optimization techniques (EAs and swarm intelligence) in addressing the optimization of the same protocol (Alba et al., 2007a; Alba et al., 2007b). Some other authors have improved the search of optimized DFCN parameterizations by applying parallel NC algorithms (Durillo et al., 2008; Segura et al., 2009). All these studies faced the optimization of the reliability, while reducing the network utilization and the delay of the operation.

Probabilistic broadcasting has been also analyzed to be optimized by using multi-objective NC. The study by Abdou et al. (2011) evaluated two multi-objective EAs to optimize communication parameters according to a given node density in the MANET. Later, the same author proposed the *Autonomic Dissemination Method* (ADM) specifically designed for VANETs, which takes into account both, the network density and the priority of the beacons to be broadcasted.

In Chapter 5 of our thesis, we have defined specific adaptive beacon broadcasting protocols for vehicular communications with efficient congestion control. These algorithms are based on different distributed greedy algorithms.

3.3.3 Natural Computing for Hardware Platform Deployment

Important agents of the VANET architecture are the RSUs. As Section 2.2.1 described, they have two main functions: **1)** acting as information sources for some applications and **2)** extending the VANET connectivity. Thus, in the current literature different studies are addressing the design of such a infrastructure as an optimization problem, the *roadside deployment problem* (RSU-DP). The RSU-DP consist in placing a set of RSUs along (the roads of) a given area, maximizing the network capabilities and minimizing the deployment costs

Most of these works analyze RSU-DP as a extended version of the *Radio Network Design* (RND) problem (Vega et al., 2007). However, as most nodes in VANETs are vehicles, the design of the roadside platform prioritizes locations taking into account road traffic information as speed of the vehicles, traffic density, etc. Exact methods, heuristics, and some metaheuristics have been utilized to solve the RSU-DP and related problems in the literature.

Aslam et al. (2012) applied the *Balloon Expansion Heuristic* (BEH) and *Binary Integer Programming* (BIP) to minimize the reporting time installing a fix number of RSUs in Miami, USA. They utilized information relative to the speed, traffic density, and likelihood of incidents for the computations. BEH performed better than BIP in the reported experiments. A *Voronoi*-based algorithm was applied to optimize packet loss, communications delays, and network coverage, while minimizing the number of RSUs required in Nashville, USA (Patil and Gokhale, 2013). This approach used information about the speed of vehicles and the traffic density to evaluate the solutions. Trullols et al. (2010) defined the *Maximum Coverage with Time Threshold Problem* (MCTTP) to maximize the number of vehicles that get in contact with the RSUs for a giving amount of time over the considered area and given a number of RSUs. Three different greedy algorithms with different knowledge of the road topology and identity of the vehicles are proposed. These approaches were applied taking into account real road and mobility data from Zurich, Switzerland. The results showed that knowledge of vehicular mobility is the main factor in achieving an optimal roadside deployment. Finally, Ben Brahim et al. (2014) applied specific versions of the *PageRank* (Langville and Meyer, 2011) and of the *0-1 Knapsack* problem solvers to select the optimal set of RSU positions within the cost range in Doha (Qatar). The algorithms took into account real traffic information data (that include traffic density, probability of hazardous situations, etc.) to compute the importance of potential locations. Knapsack algorithm showed better performance when the available budget is not highly restricted.

Just few studies have analyzed NC to deal with this optimization problem. They have been utilized to obtain accurate solutions within reasonable computational costs. An early approach joined GA and a VANET simulator to optimize the QoS of the communications in a given area of Brunswick, Germany (Lochert et al., 2008). Other two studies applied the same NC algorithm to optimize the coverage. Cavalcante et al. (2012) compared GA against the greedy approach proposed in Trullols et al. (2010), showing that the GA solutions obtained better vehicle coverage than those given by the greedy approach. Cheng et al. (2013) used just geometry-based coverage information about the roads (without vehicles mobility related data) of Yukon Territory (Canada). In this study, GA outperformed the *α–coverage* algorithm, which consists in placing the RSUs in the center of the junctions.

All this research analyzed a problem that is essentially multi-objective (maximizing QoS while minimizing deployment cost) by using mono-objective NC algorithms. This drove us to accomplish the study presented in Chapter 6, which address RSU-DP applying an explicit multi-objective formulation. In this manner, the VANET designers may have a set of accurate solutions, which present different trade-off between QoS and cost, to efficiently decide the design to be deployed.

3.4 Evaluation of the Results

In this research work, we have addressed real-world problems related to vehicular networks. Therefore, the desirable manner to evaluate the results is to validate them in real-world environments (by performing outdoor testbeds or realistic VANET simulations), which has been one of the regular practices carried out in here. Nevertheless, we have also wanted to evaluate NC in addressing such kind of problems, in order to provide a powerful base tool to be used in further VANET optimization problems. Thus, in this section we specify how we carried out the NC evaluation.

The natural inspired methods applied in this thesis are non-deterministic, hence different executions of the same algorithm over the same problem instance can produce different results. This can cause inconveniences to researchers when evaluating those results, and in the comparison of different algorithms. Besides, in the domain of optimization problems with non-deterministic algorithms is commonly adopted the comparisons on the basis of empirical data. For these reasons, some methodology based on well defined indicators should be established. In this sense, there are two types of indicators: the ones to evaluate the quality of the computed solutions and those used to measure the performance in terms of required computation time or the amount of resources they use. There are different specific quality indicators depending on whether the problem is mono-objective or multi-objective. Let us to describe them, afterwards we will present the performance metrics used as well as the statistical analysis procedure adopted in this thesis.

3.4.1 Quality Indicators

The optimization problems treated in this PhD thesis have not a known the optimum value beforehand (as most of real-world problems), and therefore, we cannot use *hit rate* metric (the ratio of times that the optimum is obtained). Thus, a most commonly adopted approach is to finish the algorithms after a given computational effort has been spent (like visiting a maximum number of points of the search space or running for a given time), and then, evaluate the quality of the solutions obtained.

In **mono-objective optimization**, different metrics about the final fitness computed are used, such as *average*, *standard deviation*, *median*, *maximum*, *minimum* values obtained after a given number of independent runs. It is commonly adopted the use of 30 executions as the minimum accepted number of runs, though higher values (such as 100) are recommended (this is very dependent on the run time of the algorithms).

Multi-objective optimization algorithms compute an approximation set to the optimal Pareto front (see Section 3.1.1), which stores a set of non-dominated solutions that presents different trade-offs between the objectives. In general, two properties are usually evaluated in MOP: convergence and a uniform diversity. In this thesis the three major performance used metrics are utilized to evaluate the multi-objective NC approaches:

- *Hypervolume* (I_{HV}) (Zitzler and Thiele, 1999): This indicator calculates the volume, in the objective space, covered by members of a non-dominated set of solutions Q. Formally, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as its diagonal corners. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its I_{HV} is calculated:

$$I_{HV} = volume\left(\bigcup_{i=1}^{|Q|} v_i\right) \tag{3.10}$$

- *Epsilon* (I_ϵ) (Knowles et al., 2006): This indicator measures the smallest distance one would need to to translate every solution in a given front A so that it dominates the optimal Pareto front of the problem. Mathematically, given $\vec{z^1} = \left(z_1^1, ..., z_n^1\right)$ and $\vec{z^2} = \left(z_1^2, ..., z_n^2\right)$, where n is the number of objectives:

$$I^1_{\epsilon+}(A) = inf\left\{\epsilon \in \mathbb{R} \mid \forall \vec{z^2} \in PF^* \exists \vec{z^1} \in A : \vec{z^1} \prec_\epsilon \vec{z^2}\right\} \tag{3.11}$$

  where $\vec{z^1} \prec_\epsilon \vec{z^2}$ if and only if $\forall 1 \leq i \leq n : \vec{z^1} < \epsilon \vec{z^2}$.

- *Spread* (I_Δ) (Deb, 2001): This metric evaluates the extent of spread by the set of computed solutions and it is defined as:

$$I_\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left|d_i - \bar{d}\right|}{d_f + d_l + (N-1)\bar{d}} \tag{3.12}$$

  where d_i is the *Euclidean distance* between consecutive solutions, $\bar{d}$ is the average of these distances, and d_f and d_l are the Euclidean distances to the extreme solutions of the optimal Pareto front in the objective space. This indicator takes a zero value for an ideal distribution, pointing out a perfect spread of the solutions in the Pareto front.

- *Generational Distance* (*GD*): This metric measures how far the elements in the computed front are from those in the optimal Pareto (Veldhuizen and Lamont, 1998) and it is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{3.13}$$

  where N is the number of solutions in the approximated front and d_i is the *Euclidean distance* (measured in objective space) between each of these solutions and the nearest member in the optimal Pareto front. A value of *GD*=0 indicates that all the generated elements are in the Pareto front.

After presenting the quality indicators for both, mono-objective and multi-objective optimization, we will define the performance indicators utilized to asses our algorithms.

3.4.2 Performance Indicators

The performance evaluation is carried out by measuring the amount of computational resources used by the algorithm (computational effort), which are generally calculated as the *computation time* or as the *number of solutions visited* in the search space. It is widespread among the research community the combined use of both metrics in order to obtain realistic picture of the computational effort. Therefore, we have followed this advice in our research.

We discuss here the main indicators used in the literature to evaluate the performance of parallel algorithms because this thesis includes studies applying parallel bio-inspired methods. Among all the metrics used, the most common ones used by the research community are the *speedup* and the *efficiency*.

The speedup evaluates how much faster a parallel algorithm is than its corresponding sequential version. It is computed as the ratio of the execution times of the sequential algorithm (T_1) and the parallel version executed on m computing elements (T_m) (Equation 3.14). When applied to non-deterministic algorithms, such as the NC ones applied in our works, the speedup should compare the *mean* values of the sequential and parallel execution times (Equation 3.15) (Alba and Tomassini, 2002). The ideal case for a parallel algorithm is to achieve linear speedup ($S_m = m$), but the most common situation is to achieve sublinear speedup ($S_m < m$), mainly due to the times required to communicate and synchronize the parallel processes.

The efficiency is the normalized value of the speedup, regarding the number of computing elements used to execute a parallel algorithm (Equation 3.16). This metric allows the comparison of algorithms eventually executed in non-identical computing platforms. The linear speedup corresponds to $e_m = 1$, and in the most usual situations $e_m < 1$.

$$S_m = \frac{T_1}{T_m} \quad (3.14) \qquad S_m = \frac{E[T_1]}{E[T_m]} \quad (3.15) \qquad e_m = \frac{S_m}{m} \quad (3.16)$$

3.4.3 Statistical Analysis of the Results

After the definition of the indicators of quality and performance, we now present the statistical analysis performed here to extract correct conclusions from the results. In this sense, a number of independent runs are carried out to obtain a set of values for each indicator. From the statistics viewpoint, these data can be considered as a sample from a probability density function, and therefore, they can be compared by means of statistical tests (Demšar, 2006; Sheskin, 2007), which are used to validate and to provide confidence to our empirical analysis .

The procedure adopted in our research work is as follows (see Figure 3.4). First, a *Kolmogorov-Smirnov* statistical test is carried out to check whether the samples are normally distributed (*Gaussian*) or not. For normal distributions, the homoskedasticity (i.e., equality of variances) is checked using the *Levene* test. If the Levene test returns a positive value, an *ANOVA* test is performed; otherwise a *Welch* test is performed. For non-normal distributions, *non-parametric test* are performed,i.e., *Kruskal-Wallis* or *Wilcoxon* test (to compare two distributions) or *Friedman Rank* statistical test (to compare/rank more than two distributions). If Friedman test returns the rank with statistical confidence, *post hoc* statistical test can be performed by using Wilcoxon or *Holm* test to confirm the Friedman results.

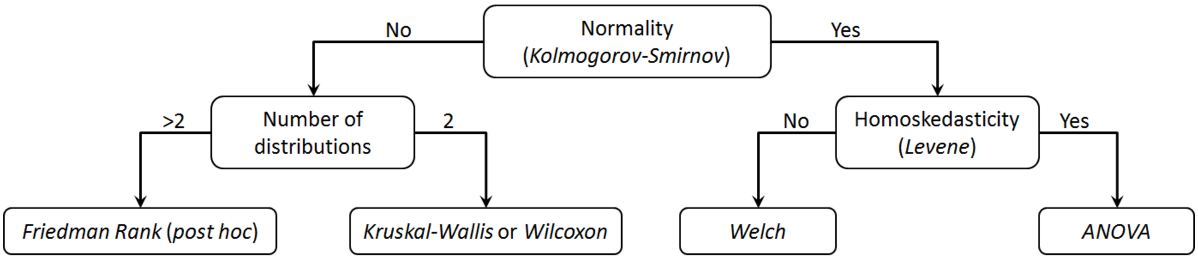


FIGURE 3.4: Statistical validation analysis process of the experimental results.

Part II:

Optimization and Experimentation in Vehicular Networks

The dictionary is the only place that success comes before work. Work is the key to success, and hard work can help you accomplish anything.

Vince Lombardi

Chapter 4

Off-line Optimization of Vehicular Communications

THIS chapter analyzes the use of NC to automatically search for high-quality parameter settings of VANET protocols previous to its deployment. The search is performed by an off-line optimization process that couples NC with a VANET simulation procedure. This methodology is applied to improve the QoS of a file transfer and two routing protocols, as well as we analyze the optimization process of the routing energy-efficiency. In addition, we study different variations of the optimization algorithms to improve the performance of NC in solving off-line protocol optimization: extending the basic operators and devising new ones, parallelizing the used techniques, and defining multi-objective models of the problem.

4.1 Introduction

VANETs present a set of special characteristics that negatively affect the quality of the communications (see Section 2.4), and therefore, that limit the accuracy of their applications. This could expose road users to hazardous situations (Benslimane, 2004). Thus, data dissemination in vehicular environments is a critical issue in today's research. Hence, the research community is very active with hot topics, creating new protocols and improving the existent ones (Chen et al., 2011; Ding et al., 2011; Dua et al., 2014; Lee et al., 2010; Santa et al., 2009).

A promising research line proposes the modification of competitive MANET communication protocols to adapt them to the special case of VANETs. One way to modify the protocols operation is by changing the values of the configuration parameters that govern them (timers, counters, etc.). This approach basically keeps the base same protocols but adapts them to new working environments. In this chapter, we propose the idea of improving the protocols software operation in VANETs by optimizing their configuration parameters.

The protocol's configuration parameters have a strongly non-linear relationship with each other and a complex influence on the final performance. In fact, they represent a mix of discrete plus continuous variables which makes it a hard challenge to find the *best* configuration in a real world vehicular environments. The performance evaluation of each protocol configuration requires a high number of VANET simulations that take in order of minutes each one. Thus, exact and enumerative methods are not applicable for solving the underlying optimization problem of finding the best configuration of a given protocol, because they require critically long execution times to perform the search, and because we are far from having a traditional analytic equation representing the protocol (to later optimize it). In this context NC is a promising approach to find accurate efficient protocol configurations in reasonable times.

Specifically, in this PhD thesis we propose a methodology to optimize VANET protocols by using NC. The procedure adopted in our analysis is summarized as follows:

1. analyzing the protocol operation and performance in VANETs in order to study the configuration parameters to be improved,
2. defining an optimization problem to automatically tune the protocol parameters according to all the possible feasible parameter configurations (search space) and a given set of performance metrics (e.g., QoS) to define the objective (fitness) function,
3. selecting the NC algorithms (as efficient search engines) that better fit the requirements of the defined optimization problem,
4. generating realistic VANET instances to evaluate the computed solutions/protocol configurations (simulating them and obtaining the performance metric values),
5. running the NC algorithms to compute accurate VANET protocol configurations that improve its performance in vehicular environments,
6. and comparing the parameterizations computed by NC against the state-of-the-art ones.

Focusing on the optimization part, NC techniques are coupled with a a realistic VANET simulator, which is utilized to accurately evaluate the solutions (protocol parameterizations) computed by the NC algorithm. As shown in Figure 4.1, during the protocol optimization the NC algorithm performs the search operations and invokes the simulation process (VANET evaluation). After the simulation, the results are used to compute the fitness function that guides the search.

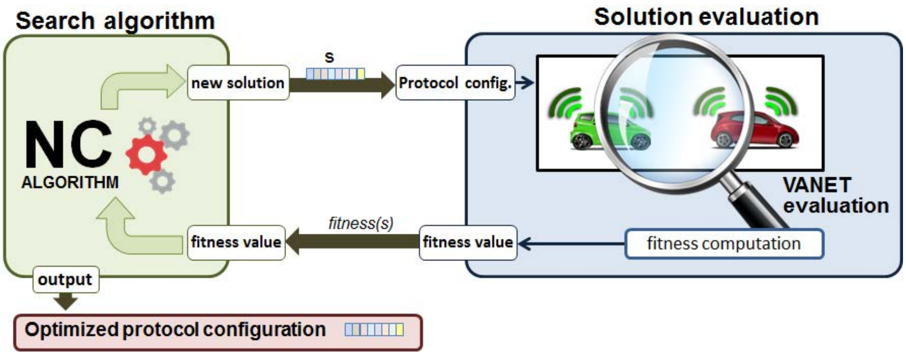


FIGURE 4.1: Optimization strategy to optimize VANET protocols.

Currently, OBUs are not highly energy constrained because they benefit from the vehicle's electricity. Nevertheless, VANETs can involve other wireless devices such as smartphones, roadside units, and sensors, that are fed with batteries or other energy sources. Limiting the energy consumption in vehicular communications may help the use of *green* renewable energy sources as solar cells installed in the used wireless devices.

A similar strategy utilized to improve the QoS of the VANET communication has been applied to reduce their energy consumption, thus finding energy-efficient protocol configurations. Therefore, in this chapter we address two main important issues in vehicular communications: the optimization of the QoS and the reduction of the energy consumption of the protocols. In other systems, this is an improvement or a twist for efficiency; however, in VANETs it is a mandatory step to get the communications actually working.

In short, this chapter analyzes the **optimization of the QoS** of a **file transfer** and two types of **routing protocols**. We have selected two different types of routing protocols (a proactive one and a reactive one), which represent two different ways of operating, in order to show the robustness of the use of NC in this filed. In addition, it studies the **energy-efficiency** optimization of a routing protocol.

The chapter is organized as follows: Section 4.2 applies NC to the optimization of the QoS a file transfer protocol. Section 4.3 analyzes the use of NC on the optimization of a proactive routing software. Section 4.4 concentrates in how to reduce the energy consumption routing data by using a parallel NC algorithm. Section 4.5 improves the QoS of a reactive routing protocol by applying parallel multi-objective NC techniques. Finally, Section 4.6 offers a global vision on applying off-line optimization in VANETs. We now turn to present the common definitions required for all subsequent work of this chapter.

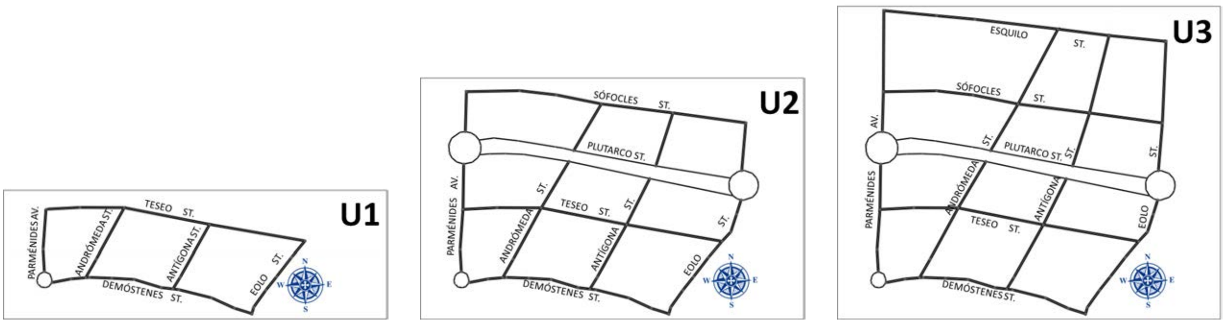
4.1.1 Instances for the Evaluation of VANET Communications

The evaluation of vehicular communications is a major concern as it has already been discussed in Section 2.6. In this thesis it has been made a great effort to defined realistic VANET simulations in order to obtain accurate results, i.e., as close as possible to the real world communications.

The simulation of communications used in our studies comprises the utilization of a widely used network simulator, the *Network Simulator* (two versions are used *ns-2* and *ns-3*). This network simulator is combined with one of the following two road traffic simulators (mobility models generators) to define the movement of the nodes as real vehicles: the VanetMobiSim (Härri et al., 2011) and the Simulation of Urban MObility (SUMO) (Krajzewicz et al., 2012). The main advantage of employing traffic simulators is that they can be used to generate realistic VANET environments by automatically selecting real areas from freely available digital maps, e.g., OpenStreetMap (Haklay and Weber, 2008), taking into account real road directions, traffic lights and signs, etc.

This section introduces the main characteristics of the road VANET instances used in the simulations performed in the studies presented in chapter. The off-line optimization take into account both, **urban** (metropolitan) and **highway** areas, because the vehicular communications behave quite differently in this two types of roads.

For the **urban scenarios**, three different geographical area sizes have been selected from the downtown of Málaga, in Spain, to study the scalability of the approaches proposed. The three urban areas are the U1, the U2, and the U3 that cover areas of 120,000 m^2, 240,000 m^2, and 360,000 m^2, respectively (see Figure 4.2). Additionally, the analysis is extended by studying how do various road traffic densities affect the protocols' performance.



a) *U1* area. b) *U2* area. c) *U3* area.

FIGURE 4.2: Road maps used to define the urban VANET instance.

The **highway instance** covers a stretch of road of one kilometer with four lanes and two directions (two lanes per direction) without buildings and semaphores. In this case, the absence of obstacles is made up for the handicap of the high speed of vehicles, which also interferes the communication among vehicles.

4.1.2 Metrics Used to Evaluate VANET Protocols

Our off-line protocol optimization targets at optimizing the QoS (performance of the protocol) and/or the energy-efficiency of a given software protocol. Therefore, the solutions have to be evaluated in terms of these objectives.

The QoS Metrics Evaluated

Two different types of protocols will be optimized in terms of QoS (file transfer and routing protocols), and therefore, the evaluation of the QoS is performed differently depending on the type of protocol evaluated.

The QoS of a file transfer protocol is measured according to four metrics (García-Nieto et al., 2010):

- *Transmission time*: Differences of the times between request and complete reception of a file.
- *Number of lost packets*: Number of the data segments or chunks generated by the file transfer protocol which have been lost during the file transfer.
- *Total data transferred*: Amount of units of data, such as bits or bytes, transferred from a source node and correctly received by a destination node.
- *Effective data rate*: Amount of units of data transferred per unit time from a source node and correctly received by a destination node.

The QoS of the routing protocols can be evaluated by taking into account these four metrics (Das et al., 2000):

- *Packet delivery ratio (PDR)*: Fraction of the data packets originated by an application that the routing protocol delivers correctly to the destination node.
- *Normalized routing load (NRL)*: Ratio of administrative routing packet transmissions to packets delivered.
- *End-to-end delay (E2ED)*: Difference between the time the data packet is originated by an application and the time this packet is received at its destination.
- *Routing path length (RPL)*: Number of hops packets take to reach their sink nodes.

Power Consumption Evaluation

The energy required for each device to perform the communications depends on its communication mode, which are: **a)** *idle*, the default state of wireless interfaces in ad hoc networks, in which nodes keep listening and the interface can change the state and start transmitting or receiving packets; **b)** *transmit* and **c)** *receive* states, when the nodes are sending and receiving data through the medium, respectively; and **d)** *sleep* state is when the node radio is turned off, and thus the node is not capable of detecting any signal. The studies carried out in this thesis deal with the optimization of the power consumption of the two operational states that act during the packet exchange: transmit and receive states, the most expensive ones in terms of energy consumption. Therefore, we consider the *per-packet power consumption* by Cano and Manzoni (2000).

The energy is computed according to the power requirements in transmitting (P_{send}) and receiving (P_{recv}) states, and the time needed to transmit the packets ($time$). These values are obtained by using the network interface card (NIC) characteristics of electric current (I_{send}, I_{recv}) and power supply (V_{send}, V_{recv}) in each state, the size of the packets ($PacketSize$), and the bandwidth ($Bandwidth$). Equations 4.1 and 4.2 represent the energy required for packet transmission (E_{send}) and for packet reception (E_{recv}).

$$E_{send} = P_{send} \times time = (I_{send} \times V_{send}) \times \frac{PacketSize}{Bandwidth} \tag{4.1}$$

$$E_{recv} = P_{recv} \times time = (I_{recv} \times V_{recv}) \times \frac{PacketSize}{Bandwidth} \tag{4.2}$$

In order to compute realist results in the simulations carried out to evaluate the communications, the real IEEE 802.11p *Unex DCMA-86P2 NIC* (Unex, 2015) is modeled as wireless interface of the VANET nodes. According to the specification of this transceiver, the power consumption is from 440 milliamperes (mA) in transmitting mode, and from 260 mA in receiving mode, and it is fed with 5.0 Volts. This NIC uses a 6 Mbps bandwidth implementation of the standard IEEE 802.11p. Thus, the power consumption in transmitting (E_{send}) and receiving states (E_{recv}), in Joules, are given by Equations 4.3 and 4.4, respectively, where the packet size is given in bits.

$$E_{send} = (440 \times 5) \times \frac{PacketSize}{6 \times 10^6} \tag{4.3}$$

$$E_{recv} = (260 \times 5) \times \frac{PacketSize}{6 \times 10^6} \tag{4.4}$$

The total power consumption for a packet transmission is the sum of the costs incurred by the sending node and all receivers, whether they are the destination nodes or not. Equation 4.5 computes the total power consumption per packet (E_{total}) when there are r receiver nodes in the communication range of the sender.

$$E_{total} = E_{send} + \sum_{i=1}^{r} E_{recv} \tag{4.5}$$

The **energy gap** (GAP_{energy}) is computed to represent the percentage of energy saved by using some parameter configuration (c) regarding the energy consumption of standard RFC (see Equation 4.6).

$$GAP_{energy}(c) = \frac{E_{total}(RFC) - E_{total}(c)}{E_{total}(RFC)} \times 100 \tag{4.6}$$

4.1.3 Specific NC Operators for VANET Protocol Optimization

The NC algorithms applied in García-Nieto et al. (2010) and in Toutouh et al. (2012b) offered very competitive results in off-line protocol optimization. However, the algorithms, which utilized canonical operators, suffered from low population diversity and early stagnation and from the generation of pointless solutions. For this reason, we have defined *initialization*, *crossover*, and *mutation* operators by introducing problem-related information, which have been used in different studies carried out in this thesis. Following, the *diagonal uniform initialization*, the *functional OLSR crossover*, the *OLSR-μ* and *AODV-μ mutation* operators are defined.

Diagonal Uniform Initialization

The diagonal uniform initialization defined in our thesis distributes the solutions of the solution set over different areas of the search space (Toutouh and Alba, 2012c). The initialization operator splits the search space into $solset\_size$ (solution set size) diagonal subspaces, and it locates each solution in each subspace. Equation 4.7 summarizes the procedure.

$$s_{k,i}^{(0)} = z_{(i,MIN)} + \rho^k \qquad i \in [0, n\_par - 1], k \in [0, solset\_size - 1] \tag{4.7}$$

In Equation (4.7), $s_{k,i}^{(0)}$ is the initial value for each component (protocol parameter) i in the *solution vector* of the k-th solution, set according to a *seed* $z_{(i,MIN)}$, and a randomly distributed value ρ^k computed by using the diagonal subspaces limits and a random value $\alpha \in [0, 1]$ (see Equation 4.8). The $z_{(i,MAX)}$ and $z_{(i,MIN)}$ are the upper and lower range values for the i-th parameter of the optimized protocol that has n_par configuration parameters, respectively.

$$\rho^k = \left(\frac{k+\alpha}{solset\_size}\right) \times (z_{(i,MAX)} - z_{(i,MIN)}) \tag{4.8}$$

OLSR-μ Mutation Operator

After analyzing the algorithm of the OLSR protocol (Clausen and Jacquet, 2003), a new mutation operator is defined in Toutouh et al. (2013). This mutation modifies simultaneously just the genes that encode OLSR related parameters, i.e., the parameters that together control a given protocol procedure, but using different policies and following the OLSR optimization problem specifications. According to this idea, the mutation operator offers 22 different movements in the solution space. For example, HELLO_INTERVAL ($x_{p,0}^{(g)}$) and NEIGHB_HOLD_TIME ($x_{p,4}^{(g)}$) parameters are mutated at the same time by applying Equation 4.9.

$$\begin{aligned} x_{p,0}^{(g+1)} &= \beta_0 \times (z_{(0,MAX)} - z_{(0,MIN)}) \qquad \beta_0 \in [0,1] \\ x_{p,4}^{(g+1)} &= \beta_4 \times (z_{(4,MAX)} - z_{(4,MIN)}) \qquad \beta_4 \in [0,1] \end{aligned} \tag{4.9}$$

AODV-μ Mutation Operator

The AODV-μ mutation operator is introduced in Toutouh and Alba (2015c) This mutation keeps the solutions in the values of correct operation of the AODV protocol (Perkins et al., 2003). The movements are limited by the lower and the upper values of the parameter ranges.

The mutation operator introduces new randomly generated information, and therefore, diversity to the population/swarm of the pMOAs. However, the AODV-μ mutation values are limited by the lower and upper values of the parameter ranges (see Equation 4.11). Equation 4.10 defines the values that depend on a uniform randomly distributed value $\beta_i \in [-0.5, 0.5]$ and on the range of values of the i-th parameter of AODV.

$$new\_s_{k,i} = s_{k,i}^{(g)} + \beta_i \times (z_{(i,MAX)} - z_{(i,MIN)}) \tag{4.10}$$

In Equation 4.11, $s_{k,i}^{(g+1)}$ is the new value computed for a mutated parameter i of the k-th solution set to $new\_s_{k,i}$ according to Equation 4.10. If the movement does not fulfill the range restrictions, then the i-th parameter is set to the upper value of its range ($z_{(i,MAX)}$) if $new\_s_{k,i} > z_{(i,MAX)}$ or to the lower value ($z_{(i,MIN)}$) if $new\_s_{k,i} < z_{(i,MIN)}$.

$$s_{k,i}^{(g+1)} = \begin{cases} new\_s_{k,i}, & \text{if } new\_s_{k,i} \in [z_{(i,MAX)}, z_{(i,MIN)}] \\ z_{(i,MAX)}, & \text{if } new\_s_{k,i} > z_{(i,MAX)} \\ z_{(i,MIN)}, & \text{if } new\_s_{k,i} < z_{(i,MIN)} \end{cases} \tag{4.11}$$

Functional OLSR Crossover Operator

The functional OLSR crossover operator was introduced in Toutouh and Alba (2012b) and it is a modified version of *N-point crossover* operator for real-valued problem encoding. It defines a linear combination of two chromosomes (*Parent A* and *Parent B*) to generate two new individuals (*Offspring A* and *Offspring B*). In this operator, we include problem-related information to avoid pointless configurations.

Thus, we define the concept of *related-genes* to refer to those genes that represent OLSR parameters which have relation between their values according to the description of OLSR in the RFC 3626 (Clausen and Jacquet, 2003). The functional OLSR crossover operator performs the exchange of three pair of related-genes (HELLO_INTERVAL and NEIGHB_HOLD_TIME;

MID_INTERVAL and MID_HOLD_TIME; and TC_INTERVAL and TOP_HOLD_TIME) and two individual-genes (WILLINGNESS and the DUP_HOLD_TIME). Thus, we have defined five different possible gene exchanges (see Figure 4.3): the *related-genes* (*Crossover-1* to *Crossover-3*), the WILLINGNESS (*Crossover-4*), and the DUP_HOLD_TIME (*Crossover-5*). Each time that the recombination operator is called three of these five genes exchanges are randomly applied to create the final offspring.

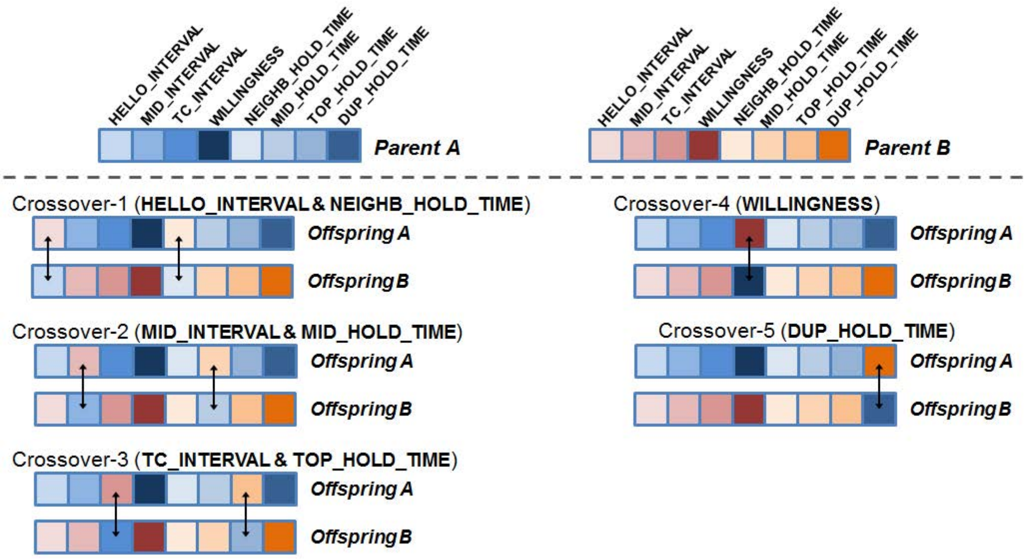


FIGURE 4.3: Different gene exchanges of the functional OLSR crossover operator.

4.2 Data Transfer Between Vehicles with Optimized QoS

File transferring is always an essential service in every communication network since their ultimate purpose is the exchange of information among the nodes, and vehicular networks are not an exception. Thus, different specific file transfer protocols have been proposed, for example: the *Vehicular Data Transfer Protocol (VDTP)* (Luna S., 2008) proposed by *CARLINK European EUREKA-CELTIC European consortium* (CARLINK, 2006), the *CarTorrent* and the *CodeTorrent* (Lee et al., 2008) designed at UCLA (University of California, Los Angeles), and the *Vehicular Information Transfer Protocol (VITP)* introduced in Dikaiakos et al. (2005). The operation of these protocols follows the same idea of splitting the files into several data blocks or chunks to be sent individually through the network. In this manner, the file transfer may be paused if there is any connection problem and it may be resumed if the nodes reconnect again.

In spite of the aforementioned protocols were specifically designed to be deployed in VANETs, their performance is limited due to the special characteristics of this kind of networks, that provoke frequent connection loss (see Section 2.4). Thus, it is desirable to optimize such protocols to provide the applications with the best file transfer service possible. Following this idea, in this thesis the ***file transfer configuration*** **(FTC)** optimization problem is defined for the first time. It consists in finding the configuration of the main parameters of a given file transferring protocol software that optimizes the VANET performance. More specifically, this section presents a study focused on applying the FTC to the VDTP protocol. For this purpose, a set of NC techniques have been analyzed in tackling this problem.

4.2.1 Vehicular Data Transfer Protocol

VDTP is a connectionless oriented protocol which operates in the application layer (Luna S., 2008). In VDTP, the communication process is carried out by two nodes, a *file petitioner*, which tries to download a given file, and a *file owner*, which initially stores the file. This transfer protocol operates by using four control packets: *FIRQ (File Information Request)*, *FIRP (File Information Reply)*, *DRQ (Data Request)*, and *DRP (Data Reply)*.

The file transfer begins when when the file petitioner sends an FIRQ to the file owner stores the file to be downloaded. The file owner replies with an FIRP, that encapsulates metadata about the file (see Figure 4.4.a). This metadata includes the information about the size of the requested file. When the file petitioner receives the FIRP, it computes the number of chunks n in which the file will be split, dividing the file size by a given block size defined in the *chunk size* protocol parameter. Then, the petitioner starts the transfer by sending the DRQ(1) packet asking for the first chunk; later, it waits for the first chunk sent in the DRP(1) packet by the owner. This operation is repeated by both, petitioner and owner, for successive chunks (exchange of DRQ(i) and DRP(i)), until the file petitioner receives the last data block in the DRP(n) packet.

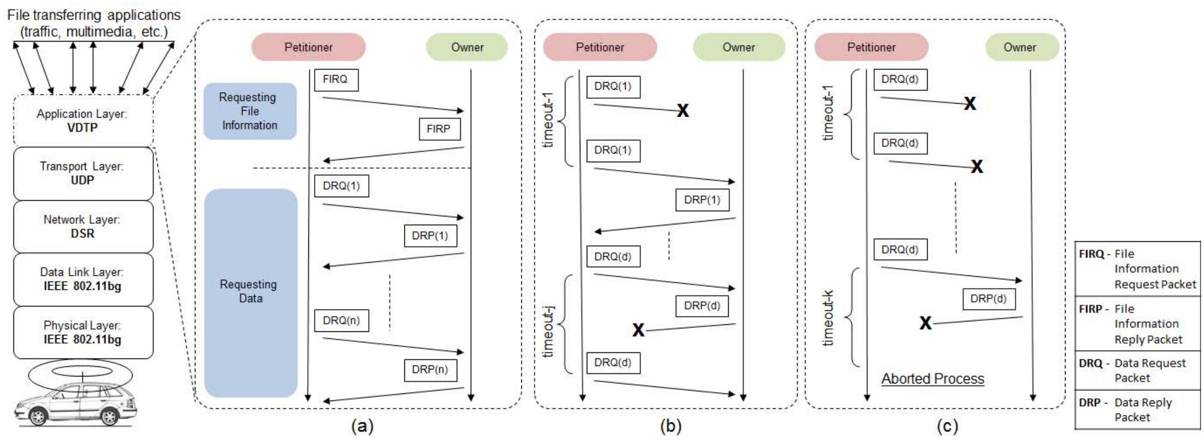


FIGURE 4.4: VDTP operation: (a) a complete file exchange is done; (b) timeout expiration and retransmission; and (c) communication refused.

Packet delivery is likely to fail during the communication process. In this sense, VDTP provides a mechanism based on timers and counters to deal with such issue. The timers control the timeout (*retransmission time*) for receiving data after sending a request packet (FIRQ or DRQ). After this timeout, the packet requested is resent to the file owner (see Figure 4.4.b). Additionally, the counters are used to control the number of times a given packet request is carried out. When the same FIRQ or DRQ packet has been repeated for a maximum number of times (*max attempts*), the file transfer is aborted (see Figure 4.4.c).

4.2.2 File Transfer Optimization Problem for VDTP

As it can be inferred from the VDTP operation, the main parameters that govern the protocol are three: the *chunk size*, the *retransmission time*, and the *max attempts*. Therefore, solving the FTC optimization problem for VDTP is finding the *best* configuration of the aforementioned parameters that optimizes the provided QoS. According to the CARLINK consortium experts, the possible values for the VDTP parameters to fulfill the requirements of VANET applications are the following ones (Luna S., 2008):

- *chunk size* is an integer value in the range of $[128 \;\; 524, 288]$ that represents the maximum quantity of data in a chuck in terms of bytes (524,288 bytes = 512 kilobytes),
- *retransmission time* (in seconds) is a real value in the range of $[1 \;\; 10]$,
- and *max attempts* is an integer value in the range of $[1 \;\; 250]$.

In this study, the QoS of VDTP is evaluated using three different metrics: the *transmission time* or communication delay, the *number of lost packets*, and the *amount of data* correctly exchanged between the nodes (see Section 4.1.2). The first metric is important because most of VANET applications have hard requirements of communication delays (see Section 2.3). The last two metrics are useful to evaluate the reliability of the protocol. Thus, solving FTC problem for VDTP consists in computing the values for each VDTP parameter that minimize both, the *transmission time* and the *number of lost packets*, while maximizing the amount *of data transferred*.

4.2.3 Implementation Details

As the FTC optimization problem on VDTP has not been analyzed in the literature, there are no previous results for comparisons. At the time of the study described here, only manually computed configurations proposed by CARLINK experts were made so far. Therefore, five NC algorithms are analyzed: four swarm/evolutionary based methods (PSO, DE, GA and ES) and a trajectory search technique (SA). These techniques are selected because they constitute a representative subset of well-known NC techniques, with suitable operators for real parameter optimization, and with heterogeneous schemes of solution set topology and evolution. This way, a set of initial results is provided in order to allow future comparisons with other optimization techniques.

Figure 4.5 illustrates the main strategy utilized in this analysis. When a given NC algorithm generates a new solution *s* it is immediately used for configuring the VDTP in a VANET simulation. This simulation evaluates the quality of the given solution by using the received VDTP configuration (the values of *chunk size*, *retransmission time*, and *total number of attempts*). After a (usually very high) simulation time, *ns-2* returns the global information about the *transmission time*, the *number of lost packets*, and the amount of *data transferred*. This information is used to compute the *fitness* function.

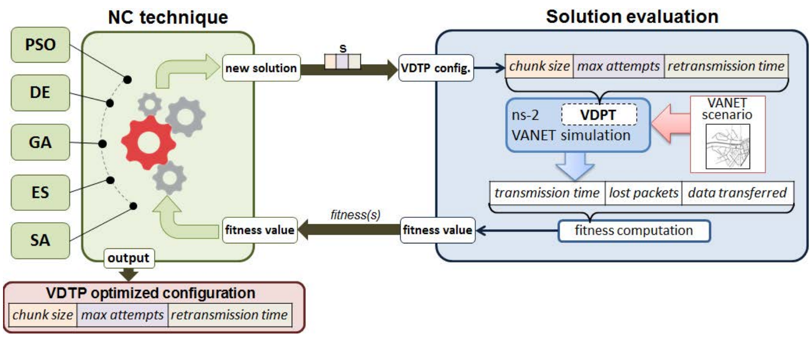


FIGURE 4.5: Optimization strategy to address FTC on VDTP.

The NC algorithms utilize the canonical operators. The problem encoding is summarized and the fitness function utilized to evaluate the solutions is below.

Problem Encoding

The VDTP protocol is governed by three configuration parameters. For this reason, each solution s is encoded as a vector of three components, each one for each parameter, i.e., $s = [chunk\_size, retransmission\_time, max\_attempts]$. The valid ranges for each one of the parameter values have already been presented in Section 4.2.2).

Fitness Function

The simulator evaluates the communications of a VANET scenario and returns the values for each QoS metric. As the vehicular communications depend on several external elements, e.g., distance between the vehicles, it is important to evaluate each solution averaging the results of different file transfers in order to provide the most accurate fitness value possible. In this study, each solution requires 10 different file transfers of the same VDTP parameterization ($N = 10$). The *fitness* value for each solution is defined as an aggregate function taking into account the three metrics ($transmission\_time$, $lost\_packets$, and $data\_transferred$), as it is shown in Equation 4.12:

$$fitness(s) = \frac{1}{N}\sum_{i=1}^{N}\frac{transmission\_time_i + lost\_packets_i}{\log(data\_transferred_i + K)} \tag{4.12}$$

In this equation, $i \in [1, 10]$ is the number of the file transfer. The data transferred is presented in logarithmic scale in order to make up for the difference in the range of values. The factor $K = 2$ avoids division by negative values or zero, preventing a possible error in the fitness calculation. Thus, the analyzed NC algorithms minimize the fitness defined in Equation 4.12 to solve the FTC optimization problem.

4.2.4 Experimental Results

This section presents the experiments carried out to solve FTC optimization problem on VDTP and discusses the main results. The NC algorithms used to optimize the QoS of VDTP are implemented using the C++ MALLBA framework (Alba et al., 2006).

Instances: VANET Scenarios

The FTC optimization problem analyzed here takes into account two different vehicular scenarios: an urban area and a highway road. Thus, we can analyze in both scenarios the behavior and performance of the compared algorithms.

The *Urban* and the *Highway* instances utilized in this study are defined by the *U1* urban area and *H* highway road, respectively (see Section 4.1.1). In both instances 30 vehicles are circulating with appropriate speeds according to the road and 20 of them are trying to send or receive files of 1024 kilobytes (KB). The communication devices of the utilized vehicles are configured using IEEE 802.11b, DSR, and UDP protocols for the PHY/MAC, routing, and transportation layers, respectively. García-Nieto et al. (2010) describes further details of the analyzed VANET instances.

Parameter Settings of the Algorithms

The five studied algorithms are configured to perform 1,000 solution evaluations per run. The swarm/population based NC techniques (PSO, DE, GA, and (μ, λ)-ES) are configured with 20 particles/individuals, performing 50 generational steps; and SA iterates 1,000 times. Table 4.1 summarizes the remaining parameters specific to each algorithm.

These parameters are selected as the most accurate after a set of initial tuning experiments. In those, a number of five combinations of parameters per algorithm and VANET instance are tested performing only 10 independent runs per combination, hence resulting in a number of 500 additional executions. However, later for the actual study we use 30 independent runs since it is the minimum to compute meaningful statistical results. Preliminary results of parameters tuning are available in Table B.1 (see Appendix B).

TABLE 4.1: Parameterization of the optimization algorithms to address FTC problem.

Algorithm	Parameter	Symbol	Value
PSO	Local Coefficient	φ_1	2.0
	Social Coefficient	φ_2	2.0
	Inertia Weigh	w	0.5
DE	Crossover Probability	Cr	0.9
	Mutation Factor	μ	0.1
GA	Crossover Probability	P_{cros}	0.8
	Mutation Probability	P_{mut}	0.2
ES	Crossover Probability	P_{cros}	0.9
	Mutation Probability	P_{mut}	0.1
SA	Temperature Decay	T	0.8

Numerical Analysis

This section discusses the results obtained by the five studied algorithms when solving the optimal FTC problem on VDTP. Table 4.2 shows the resulting fitness values regarding the *Urban* and *Highway* scenarios in terms of the average, the normalized standard deviation, the minimum (best fitness), the median, and the maximum (worst fitness) found in 30 independent runs of every algorithm.

TABLE 4.2: Final fitness values of FTC optimization for the *Urban* and *Highway* scenarios.

Instance	Algorithm	Average ± Stdev.	Minimum	Median	Maximum
Urban	PSO	**1.6346** ± 17.74 %	0.9077	**1.7809**	**1.8918**
	DE	1.7423 ± 21.33 %	**0.7389**	1.8658	2.0228
	GA	1.9086 ± 11.84 %	0.8799	1.9731	2.1614
	ES	2.1517 ± **5.88** %	1.8862	2.1222	2.4246
	SA	2.7850 ± 31.30 %	0.8730	2.1663	3.8025
Highway	PSO	**4.1761 ± 6.12 %**	3.3301	4.2513	**4.4554**
	DE	4.6631 ± 20.01 %	2.7145	4.2272	7.0531
	GA	4.3805 ± 19.85 %	**2.5345**	4.1918	5.8608
	ES	5.7833 ± 16.78 %	3.8836	6.1347	6.9421
	SA	4.4246 ± 16.73 %	3.1498	**4.0855**	5.7922

For the ***Urban* scenario**, Table 4.2 shows that PSO obtains the best result in terms of the average, median, and maximum fitness values. This result leads us to believe that using the PSO the resulting VDTP ends in an efficient communication which is fast and accurate between vehicles. Howeverm the very best VDTP configuration (minimum fitness) is found for *Urban* is reached by DE, so if the robustness is not an issue we could use DE instead PSO. ES provides the smallest deviation (5.88 %) since the results obtained are close each other, but they are far from the best obtained ones by the other NC techniques. Similar results are observed for the ***Highway* scenario**, in which PSO obtains the best average fitness value again. In terms of the minimum fitness, GA obtains the best VDTP configurations for the *Highway* scenario. For this instance the least competitive NC technique is ES, i.e., it computes the highest (undesired) average, minimum, and median fitness values.

In order to provide such comparison with statistical confidence, the Friedman and the Wilcoxon Signed Rank non-parametric statistical tests (Sheskin, 2007) are performed because the distributions violate the condition of normality required to apply parametric tests. For *Urban* instance, the algorithm that significantly obtains the best results is PSO according to Friedman and Wilcoxon tests. The second and third ranked algorithms by Friedman are DE and GA, respectively. For *Highway* instance, there is not a clear trend about which one is statistically the algorithm that performs the best since Friedman ranked SA, GA, and PSO as first, second, and third best algorithms, but these three algorithms do not show statistical differences with each other. These statistical results lead us to think that, in spite of the global (robust, predicable) competitive behavior of PSO, the different requirements implicit to both instances implies that each algorithm can show quite different results depending on the VANET scenario on which it operates. The results of these statistical tests are detailed in tables B.3, B.4, and B.5 (see Appendix B).

Algorithms Performance Analysis

This section basically lies in analyzing the quality of solutions during the whole evolution process of a given NC technique. Figure 4.6 illustrates the graphs of the best fitness values obtained through the median execution in *Urban* and *Highway* instances.

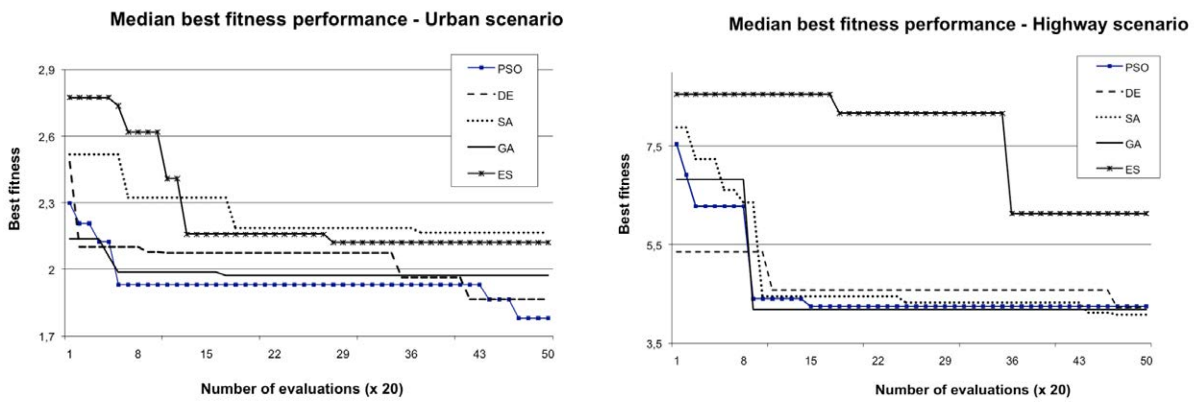


FIGURE 4.6: Best fitness evolution for the median run when solving FTC problem.

In both figures PSO and DE tend to converge in the same range of solution evaluations, and they could improve their fitness even in the final steps of the evolution process. GA shows a similar trend but it is subjected to an early stagnation. Finally, the high variability behaviors observed in ES, and specifically in SA, for both instances confirm us the high dependency of such algorithms to each different VANET scenarios (they do not seem robust in this application).

Execution Time Analysis

Concerning the run time each algorithm spends in the experiments, Table 4.3 shows both the average time in which the best solution is found T_{best}, and the total average execution time per run T_{run}. In general, SA shows the shortest times to find its best solution for the two VANET instances. This is mainly due to SA quickly falls in local optima hence obtaining weak results in *Urban* scenario. Nevertheless, this behavior can be an advantage for *Highway* scenario where SA obtained accurate solutions with a fast performance. Besides this algorithm requires less internal operations. As expected in PSO and DE, they spent close executions times for the two VANET instances since they have similar internal operations. This resemblance is also registered in the two evolutionary algorithms, GA and ES.

As a summary, the algorithms use between 80 and 150 minutes for the Urban scenario, and between 23 and 60 minutes for Highway scenario. This global low effort in the protocol design is completely justified by the subsequent benefits obtained in the global data transmission time and loss of packets.

TABLE 4.3: Average execution times (seconds) per independent run of each algorithm in solving FTC.

Instance	Algorithm	T_{best} (seconds)	T_{run} (seconds)
Urban	PSO	4.68E+03	7.95E+03
	DE	4.37E+03	7.12E+03
	GA	3.48E+03	6.68E+03
	ES	5.46E+03	9.00E+03
	SA	**2.18E+03**	**4.76E+03**
Highway	PSO	1.39E+03	2.19E+03
	DE	9.82E+02	2.10E+03
	GA	8.83E+02	1.56E+03
	ES	9.84E+02	1.47E+03
	SA	**5.85E+02**	**8.45E+02**

Scalability Analysis

In order to analyze how do various network sizes affect the performance of the NC techniques, they are executed over unseen two urban VANET instances: *U2* and *U3* (see Section 4.1.1) with 40 and 50 vehicles, respectively. Table B.2 in Appendix B presents the results of the whole scalability analysis out of 30 independent runs.

From the point of view of the fitness obtained by each algorithm, PSO keeps the best performance for the two new instances. Additionally, one of the most interesting results can be observed in GA, which arises as the second best algorithm in improving its behavior with the VANET size. Concerning to the execution time, as expected, the run times always increase with the network size.

4.2.5 Efficient VDTP Validation

After the optimization process, the configurations obtained by the NC algorithms are analyzed in terms of the QoS indicators (transmission time, number of lost packets, and amount of data transferred). Table 4.4 shows the results after simulating the best solutions found by the studied algorithms during the median run. In addition, the last row of this table contains the results of simulating the configuration of VDTP that has been used in the scope of the CARLINK project. The amount of data transferred is not shown because all the file transfers finished successfully, and therefore, for all the simulations the average data exchanged for each file transfer is 1,024 KB.

TABLE 4.4: Comparison among different VDTP configurations (improved and CARLINK experts).

Instance	Algorithm	VDTP Configuration			Simulation Results	
		chunk size	*retrans. time*	*max. attempts*	*trans. time* (secs.)	*lost packets*
Urban	PSO	41,358	10.00	3	**3.41**	**0.27**
	DE	28,278	6.00	9	3.59	0.63
	GA	31,196	3.83	9	3.61	**0.27**
	ES	23,433	10.00	8	3.50	**0.27**
	SA	19,756	6.43	3	4.22	0.36
	CARLINK	25,600	8.00	8	4.24	1.60
Highway	PSO	29,257	6.42	9	**24.67**	3.18
	DE	19,810	6.91	8	27.66	3.45
	GA	34,542	9.54	10	26.96	2.72
	ES	38,490	8.15	12	33.99	3.36
	SA	32,002	8.21	4	25.43	**2.54**
	CARLINK	25,600	10.00	10	33.08	3.27

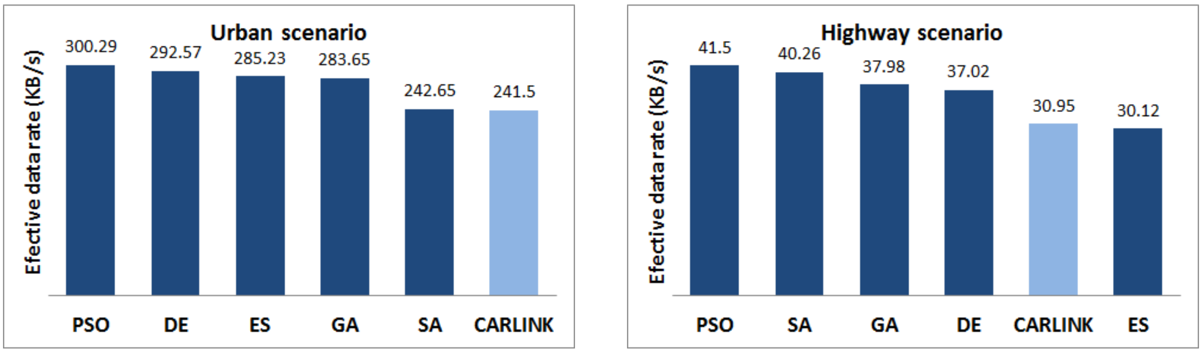


FIGURE 4.7: Final effective data rate (KB/s) of the analized VDTP configurations.

For the *Urban* scenario, the VDTP configuration obtained by PSO achieves the best performance in terms of transmission time and average number of lost packets. Specifically, in comparison to the human experts configuration of CARLINK, PSO produces a reduction in the transmission time of 0.83 seconds (19.5%) registering also a lower number of lost packets. Nevertheless, it is in the *Highway* scenario where PSO obtains the higher time reduction of 8.41 seconds (25%) regarding the human experts configuration (from 33.08 s to 24.67 s). It is noticeable that, in spite that PSO achieves a higher reduction in the transmission time than SA and GA, the fact of losing more packets (3.18 in PSO, 2.71 in GA, and 2.54 in SA) in the global transference leads SA and GA to calculate a better fitness value (as shown in Table 4.2).

A final analysis is done concerning the *effective transmission data rate* achieved. As we can see in Figure 4.7, the VDTP configuration obtained by practically all algorithms in the two VANET scenarios obtained higher effective data rates than the human configured VDTP. This clearly claims for the utilization of NC to help network designers. The actual correction of effective data rates between cars are in the order of tens of KB per second, so our savings (58.79 KB/s in Urban and 10.5 KB/s in Highway) are truly meaningful in current real applications.

As a final comment, concerning to the *effective transmission data rate* observed in Figure 4.7, practically all the configurations obtained by the NC algorithms in the two VANET scenarios outperform the human configured VDTP. This clearly claims for the utilization of NC to help network designers. The actual correction of effective data rates are in the order of tenths of KB per second, which are truly meaningful in current real VANET applications.

4.2.6 General Discussion on File Transfer QoS Optimization

This study demonstrates that the use of NC is a promising tool for the off-line optimization of the file transfer protocols in VANETs. PSO and GA provide competitive performance in finding accurate VDTP parameterizations for both, metropolitan and highway roads. These results keep competitive even when the simulated VANET grows, i.e., increasing the complexity of the vehicular environment.

From the point of view of its real world utilization, PSO computed VDTP parameterizations reduce **19%** of the transmission time in urban and **25.43%** in highway with regards to human expert configuration of CARLINK, while transmitting the same amount of data. The highest effective data rates are obtained by PSO, which are **300.39 kBytes/s** in urban VANETs and **41.50 kBytes/s** in highway roads in comparison with 241.5 kBytes/s and 30.95 kBytes/s of human experts. The results lead us to advise the final use of our automatic design NC tool. These improvements in the protocol performance have been later confirmed in **real world experiments** by using real vehicles in an open road presented in Toutouh and Alba (2011c) (see Section 7.2).

4.3 Optimization of the QoS of Proactive Routing

As it is introduced in Section 2.4, designing efficient routing protocols for VANETs is a serious challenge due to their unique and difficult features like decentralization, high mobility, and hard delay requirements. Thus, after dealing with the optimization of a file transfer protocol in the previous section, this study analyzes the use of NC algorithms to improve the QoS of the routing protocols used to perform vehicular communications.

A new optimization problem is here defined to find efficient parameterizations of a proactive (table driven and link-state) routing protocol. Proactive protocols can be recognized because they maintain routing information even before they need this information. Each node stores paths to every node in the network. This information is generally kept in a number of *routing tables* and is periodically updated (with a given frequency or when a given destination node cannot be reached). Proactive protocols are applied in vehicular environments because they present a series of features that make them well-suited for VANETs (Huhtonen, 2004): they exhibit very competitive transmission delays (which is an important feature for VANET applications) and they adapt well to the continuous topology changes.

The main drawback of such protocols is the need of maintaining the routing tables. This process is carried out by periodically broadcasting control packets to update routing tables. This drawback is negligible for scenarios with a few nodes, but for large networks the overhead of control messages could provoke network congestion. This constraints the scalability of this type of protocols. Thus, the QoS significantly depends on the selection of its parameters, what determine the protocol operation. For example, the detection of topological changes and the network load generated by the protocol can be adjusted by changing the time interval for broadcasting HELLO messages.

In the present work, we aim at defining and solving an off-line optimization problem to efficiently and automatically tune OLSR (Clausen and Jacquet, 2003), a widely used mobile ad hoc network proactive routing protocol, in order to optimize its QoS when it is used in vehicular environments. As shown in the results presented in (Gómez et al., 2005; Huang et al., 2006; Härri et al., 2006) and in the present study, OLSR admits a wide range of QoS improvement by changing the configuration parameters.

4.3.1 OLSR Routing Protocol for VANETs

OLSR is a proactive routing protocol designed for mobile ad hoc networks which show low bandwidth and high mobility. It is a version of classical link-state routing protocol, which relies in employing an efficient periodic flooding of control information using special nodes that act as *multipoint relays* (MPRs) (Nguyen and Minet, 2007). OLSR daemons running on every node of the network periodically exchange different messages in order to maintain the topology information of the entire network in the presence of mobility and failures. The core functionality is performed mainly by using three different types of messages: HELLO, TC (topology control), and MID (multiple interface declaration) messages.

- HELLO messages are exchanged between neighbors nodes (1-hop distance). They are employed to accommodate for link sensing, neighborhood detection, and MPR selection. These messages are generated periodically, containing information about the links between neighbor nodes.
- TC messages are generated periodically by MPRs to indicate which other nodes have selected it as their MPR. This information is stored in the *topology information base* of each network node which is used for routing table calculations.

- MID messages are sent by the nodes to report information about their network interfaces employed to participate in the network. Such information is needed since the nodes may have multiple interfaces with distinct addresses.

The OLSR mechanisms are regulated by a set of parameters predefined in the OLSR RFC 3626 (Clausen and Jacquet, 2003) (see Table 4.5). These parameters have been already tuned by different authors without using any automatic tool in (Gómez et al., 2005; Huang et al., 2006) and they are: the timeouts before resending HELLO, MID, and TC messages (*HELLO_INTERVAL*, *REFRESH_INTERVAL*, and *TC_INTERVAL*, respectively); the "validity time" of the information received via these message types, which are: *NEIGHB_HOLD_TIME* (HELLO), *MID_HOLD_TIME* (MID), and *TOP_HOLD_ TIME* (TC); the *WILLINGNESS* of a node to act as a MPR (to carry and forward traffic to other nodes); and *DUP_HOLD_TIME*, that represents the time during which the MPRs record information about the forwarded packets.

TABLE 4.5: Main OLSR parameters and RFC 3626 specified values.

Parameter	Type	Range	Standard configuration
HELLO_INTERVAL	$\mathbb{R}$	$[1.0, 30.0]$	2.0 s
REFRESH_INTERVAL	$\mathbb{R}$	$[1.0, 30.0]$	2.0 s
TC_INTERVAL	$\mathbb{R}$	$[1.0, 30.0]$	5.0 s
WILLINGNESS	$\mathbb{Z}$	$[0, 7]$	3
NEIGHB_HOLD_TIME	$\mathbb{R}$	$[3.0, 100.0]$	$3\times$ HELLO_INTERVAL
TOP_HOLD_TIME	$\mathbb{R}$	$[3.0, 100.0]$	$3\times$ TC_INTERVAL
MID_HOLD_TIME	$\mathbb{R}$	$[3.0, 100.0]$	$3\times$ TC_INTERVAL
DUP_HOLD_TIME	$\mathbb{R}$	$[3.0, 100.0]$	30.0 s

4.3.2 OLSR QoS optimization in VANETs

The standard configuration of OLSR offers a moderate performance when used in VANETs (Santa et al., 2009). Because of this high impact of parameters in the QoS of the protocol, an optimization problem is defined in order to discover the *best* protocol configuration. The standard OLSR parameters are defined without clear values for their ranges. The range of values each parameter can take has been defined here by following OLSR restrictions with the aim of avoiding pointless configurations.

According to that, we can use the OLSR parameters to define a solution vector of mixed integer and real variables, each one representing a given OLSR parameter. This way, the solution vector can be fine-tuned automatically by using NC with the aim of obtaining QoS efficient OLSR parameters configurations for VANETs hopefully outperforming the standard one defined in the RFC 3626. The OLSR QoS optimization problem analyzed here takes into account three metrics that evaluate the communication cost: PDR, NRL, and E2ED. Thus, the problem consist in finding the OLSR configuration that maximizes PDR and minimizes both, NRL and E2ED.

4.3.3 Implementation Details

As it happened with the precedent FTC problem, there are not previous results of optimizing the QoS of OLSR for comparisons. Therefore, four NC algorithms are analyzed: PSO, DE, GA, and SA. These techniques are selected because they craft operators for real parameter optimization and represent a subset with heterogeneous schemes for search. In addition, a random search algorithm (RAND) is studied for comparison purposes.

The main strategy used in this analysis is summarized in Figure 4.8. The NC algorithm iteratively computes solutions of (hopefully) larger QoS. The simulator evaluates the communication costs of the received OLSR parameterization (tentative solution) by using ns-2. After the simulation procedure ns-2 returns the values of PDR, NRL, and E2ED to compute the *fitness* value, and then, the algorithms can go for a new improvement iteration.

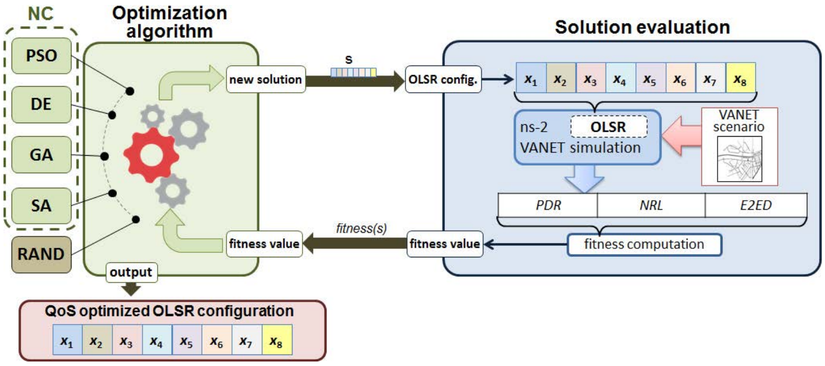


FIGURE 4.8: Strategy to address OLSR QoS optimization.

Our NC algorithms will use standard operations. Let us now explain the problem encoding and the fitness function utilized to evaluate the solutions.

Problem Encoding

As OLSR is governed by eight configuration parameters, thus the solution is encoded as a vector of eight components. The type and valid ranges for the parameter values are presented in Table 4.5. Figure 4.9 shows a representation of of this vector.

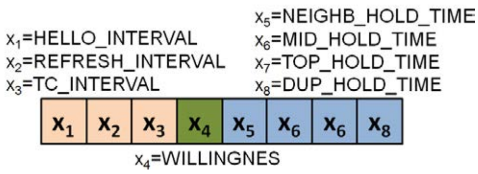


FIGURE 4.9: Solution encoding for the QoS OLSR tuning optimization problem.

Fitness Function

The simulator evaluates the communications of a given VANET under the circumstances defined by the OLSR routing parameters generated by the optimization algorithm. After the simulation, *ns-2* returns information about the PDR, NRL, and E2ED of the whole VANET. This information is used in turn to compute the fitness value of the current solution as follows:

$$fitness(s) = w_2 \cdot NRL(s) + w_3 \cdot E2ED(s) - w_1 \cdot PDR(s) \tag{4.13}$$

As previously set, the objective here consists in maximizing PDR, and minimizing both, NRL and E2ED. As expressed in Equation 4.13, an aggregative minimizing function is used, and for this reason PDR is formulated with a negative sign. In this equation, factors w_1, w_2, and w_3 are in the range [0,1] and must sum up to 1.0, with actual values 0.5, 0.2, and 0.3, respectively, in our forthcoming experimental analysis. This way, routing effectiveness (PDR) takes priority over the communication efficiency (NRL and E2ED).

4.3.4 Experimental Results

This section presents and analyzes the experiments carried out to solve optimization problem on OLSR to improve its QoS in VANETs. The NC algorithms applied to tackle the OLSR QoS optimization problem (PSO, DE, GA, and SA) are implemented in the C++ MALLBA framework.

VANET Instance for Fitness Computations

The VANET instance evaluated to compute the fitness function of the tentative OLSR configurations is defined over the previously defined U1 urban area. This scenario contains 30 vehicles moving through the roads during three minutes. The vehicles exchange data created by a constant bit rate data generator (CBR) application. Toutouh et al. (2012b) describes all the details of the used VANET.

Parameter Settings of the Algorithms

The studied five optimization algorithms are executed to reach the same stop condition (1,000 fitness function evaluations) in order to compare them. SA and RAND performs 1,000 iteration steps, and population/swarm based algorithms perform 100 generations with populations of 10 individuals/particles (100×10=1,000) each one of them. The main parameters of these algorithms are summarized in Table 4.6.

TABLE 4.6: Parameterization of the optimization algorithms to address OLSR QoS tuning problem.

Algorithm	Parameter	Symbol	Value
PSO	Local Coefficient	φ_1	2.00
	Social Coefficient	φ_2	2.00
	Inertia Weigh	w	0.50
DE	Crossover Probability	Cr	0.90
	Mutation Factor	μ	0.10
GA	Crossover Probability	P_{cros}	0.80
	Mutation Probability	P_{mut}	0.01
SA	Temperature Decay	T	0.80

Numerical Analysis

This section discusses the results obtained by the five studied algorithms when solving the QoS OLSR optimization problem. Table 4.7 shows the average and the normalized standard deviation of the fitness values obtained (out of 30 independent executions). The best (minimum), median, and worst (maximum) values are also provided.

TABLE 4.7: Results obtained in the OLSR QoS optimization.

Algorithm	Average ± Stdev.	Minimum	Median	Maximum	Friedman rank.	KW(p-value)
SA	**-0.450297±5.32%**	-0.478242	**-0.457451**	**-0.406932**	**1.40**	**3.0591E-6**
DE	-0.436897±6.86%	-0.480030	-0.435264	-0.392578	2.10	3.0660E-6
PSO	-0.432240±7.93%	**-0.482343**	-0.419734	-0.392503	2.50	3.0669E-6
GA	-0.350837±6.55%	-0.437241	-0.344612	-0.327281	4.33	3.1592E-6
RAND	-0.329878±15.16%	-0.410131	-0.329792	-0.217024	4.50	3.3579E-6

SA outperforms all other algorithms in terms of average, median, and worst fitness values. According to these measures, SA is followed by DE, PSO, and GA, respectively. Nevertheless, the best (minimum) fitness is computed by the PSO which is the algorithm that obtained the best performance in optimizing VDTP (see Section 4.2.2). Finally, as expected, the random search algorithm is the least competitive one.

With the aim of providing these comparisons with statistical confidence, the Friedman and the Kruskal-Wallis tests (Sheskin, 2007) are applied to the distributions of the results. These non-parametric tests are utilized since the resulting distributions often violate the conditions of equality of variances (heteroskedasticity) several times. The confidence level is set to 99% (*p-value*=0.01).

In effect, confirming the previous observations, the results of Friedman test ranked SA as the algorithm with the best global performance followed by DE, PSO, and GA, respectively (see sixth column of Table 4.7). Moreover, the multicompare test of Kruskal-Wallis resulted in *p-values* $\ll 0.01$ (see last column of Table 4.7). Therefore, we can claim that all the compared algorithms obtained statistically different results.

Algorithms Performance Analysis

This section presents the evolution of the quality of the solutions during the optimization process. Figure 4.10 plots the best fitness value tracked throughout the best run for each algorithm. As the figure illustrates, DE, PSO, and SA converge into the same range of solutions. But their evolution is different. SA, the best ranked algorithm, performs several gradual improvements of its solution during the whole execution.

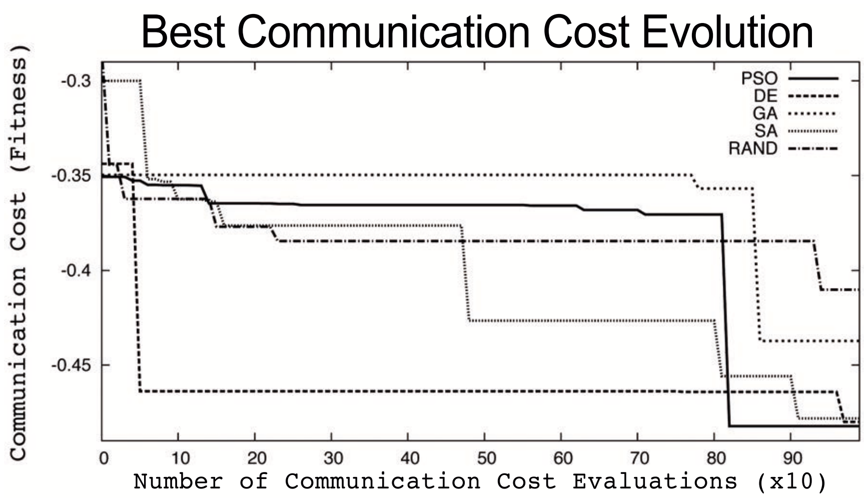


FIGURE 4.10: Best fitness evolution solving the QoS OLSR optimization problem.

Execution Time Analysis

Concerning the run time that each algorithm spent in the experiments, Table 4.8 shows the average time in which the best solution is found T_{best} and the average total run time T_{run}, which is between 12.11 and 32.66 hours. GA shows the shortest time (2.04E+04 s) to find its best solutions. It seems that this algorithm quickly falls in local optima, hence obtaining weak results (see Table 4.7). PSO needs the second shortest time (3.05E+04 s) to compute its optima. Finally, SA takes the longest time (5.78E+04 s) to find its best solutions because it performs fitness improvements during the whole execution, as Figure 4.10 illustrates. Regarding total run time (T_{run}), random search takes shorter times (4.36E+04 s) than the other algorithms since it has less internal operations. PSO is the NC technique that spends shortest mean running time (5.38E+04 s) followed by DE, SA, and GA, respectively.

TABLE 4.8: Average execution times per run of each algorithm for solving QoS OLSR optimization.

Algorithm	T_{best} (seconds)	T_{run} (seconds)
PSO	3.05E+04	5.38E+04
DE	4.29E+04	7.95E+04
GA	**2.04E+04**	1.18E+05
SA	5.78E+04	1.04E+05
RAND	3.73E+04	**4.36E+04**

4.3.5 QoS Efficient OLSR Validation

The best configurations obtained by each algorithm are compared in terms of the selected QoS indicators (PDR, NRL, and E2ED). First, they are compared with each other and with the standard proposed in RFC 3626 and the ones proposed by Gómez et al. (2005) in the VANET scenario utilized during the optimization process (Table B.6 in Appendix B shows all these configurations). Second, a further validation is performed by simulating a set of 54 different VANET scenarios applying the computed configurations. Toutouh et al. (2012b) further details the VANET instances and the experimental results.

Comparison in the VANET Used During the Optimization

Table 4.9 presents the results of simulating the urban VANET scenario used during the optimization process (U1 urban area with 30 vehicles). Columns two to four contain three human expert configurations (#1, #2, and #3) proposed by Gómez et al. (2005); column five contains the standard OLSR RFC one; and the columns six to ten show the best OLSR configurations obtained by each one of the five optimization algorithms: RAND, DE, PSO, GA, and SA.

TABLE 4.9: QoS comparisons of considered OLSR configurations.

Metric	Human experts			OLSR	Optimized configurations				
	#1	#2	#3	RFC	RAND	DE	PSO	GA	SA
PDR (%)	71.43	87.50	93.34	91.67	94.12	**100.00**	**100.00**	**100.00**	**100.00**
NRL (%)	89.54	32.48	14.13	9.52	6.93	**2.71**	2.90	13.12	4.84
E2ED (ms)	5.41	5.03	7.19	6.29	6.57	15.60	11.31	19.17	**4.73**

Examining the PDR indicator, the four NC algorithms obtain a 100% in contrast to the random search algorithm, which achieved a 94.12%. The other analyzed parameterizations perform worst. This is an important issue in highly dynamic VANETs, since a low packet delivery ratio directly implies a higher packet loss (lower reliability).

Concerning the NRL, similar results can be observed. That is, almost all the OLSR configurations computed by NC show better routing loads than the other proposals. Only GA (the worst ranked metaheuristic) obtains a NRL (13.12%) worse than the two ones obtained by the standard configuration (9.52%), although better than the three human expert configurations (#1 with 89.54%, #2 with 32.48%, and #3 with 14.13%). Reducing the routing load is important since this is a way to reduce the possibility of network failures related to congestion problem in VANETs (Wischhof and Rohling, 2005).

Finally, in terms of the E2ED, SA spends the shortest times (4.73 ms), followed by human experts configurations, standard RFC 3626, and RAND. In this case, the remaining NC algorithms (PSO, DE, and GA) show a moderate performance. Evidently, the low routing load experimented in these configurations limits the routing management operations, hence making the average E2ED worse than other configurations showing high routing load. However, it is remarkable that all the optimized OLSR parameter settings analyzed here deliver the packets within a delay shorter than 20 ms, which is the highest allowed latency for co-operative safety applications (CAMP, 2005).

Further Validation Experiments

A set of validation experiments including 54 different urban VANET scenarios are performed with the aim of validating the optimized parameters on different conditions of traffic density, network use, and area dimension. The results are evaluated in terms of four routing QoS metrics: PDR, NRL, E2ED, and RPL. Table B.7 presents the whole experimentation results computed in the simulations performed over the 54 different VANET scenarios. It has been included in the Appendix B because of the space constraints. Below, Table 4.10 summarizes these results by showing the median values for each metric and each OLSR parameterization. The best values are marked in bold.

TABLE 4.10: Median results of the validation experiments (QoS OLSR optimized configurations).

Configurations	PDR	NRL	E2ED	RPL
SA	84.76%	14.56%	**4.04 ms**	1.35
DE	84.29%	**11.98%**	10.24 ms	1.34
GA	87.85%	16.32%	4.36 ms	1.34
PSO	86.73%	12.73%	8.12 ms	1.46
RAND	88.93%	19.21%	17.16 ms	1.38
RFC	**89.56%**	23.15%	6.06 ms	**1.09**

Regarding PDR, the number of packets delivered is generally reduced with the size of the scenario. GA and SA configurations obtain the best PDR in the U1 scenario (99.95%), PSO in the U2 scenario (86%), and RFC in the U3 scenario (86.71%). Globally, the differences between the performance of all configurations in terms of this metric is just between 1% and 5%.

In terms of the routing workload generated by the protocol (NRL), the best performance is obtained by the configurations obtained by SA and DE, so they can even decrease the NRL along with the scenario size. In particular, scenario U2 seems to be a source of high routing loads since practically for all solutions (excepting the ones of SA and DE) this indicator is increased. In general, optimized OLSR configurations improve the NRL. The RFC configuration shows the highest routing load value (overall NRL=23.15%) and it is twice the one obtained by DE, which is the (smallest) best one (11.98%).

The E2ED is higher with the scenario area dimension. The OLSR configuration computed by GA required the shortest average E2ED in U1 and U2 scenarios with 2.10 ms and 3.81 ms, respectively. In scenario U3, DE obtained the best E2ED (19.19 ms). Globally speaking, the shortest median E2ED is obtained by SA with 4.04 ms.

Regarding the computed routing paths (RPL), GA obtained the shortest paths in the U1 scenario, and RFC configuration used the shortest paths in scenarios U2 and U3. In general, median path lengths obtained by the OLRS RFC required the lower number of hops. In this case, the higher frequency of routing information exchange maintains the routing tables up-to-date, although generating a higher routing load (NRL).

In summary, it can be confirmed that automatically tuned OLSR configurations by NC techniques offer the best trade-off between the four QoS metrics in the scope of the multiple scenario conditions analyzed here. Solutions obtained by NC algorithms show high rates of packets delivery (>84%), and low values of routing load (<16.5%), end to end delays (<10.3 ms), and paths lengths (<2 hops). Standard OLSR also reached accurate median values of PDR (88.9%), but with the drawback of a high routing load (>23%), which is a critical concern in this kind of networks.

4.3.6 General Discussion on Off-line Proactive Routing Optimization

The results of the experimentation carried out in this study confirm the applicability of NC algorithms in optimizing the QoS of proactive routing protocols (OLSR) in VANETs. The automatically computed configurations outperform the standard one in all the studied of VANET networks, which represent a wealth of vehicular situations.

SA outperforms the other studied algorithms when solving the defined OLSR optimization problem. However, PSO presents a competitive trade-off between the performance and the execution time requirements. Thus, it can offer accurate OLSR configurations to the experts in reasonable design times.

Concerning to the solution domain, globally, the validation experiments show that the optimized configurations generate reduced network workload, generating about **the half of the routing load** than the standard OLSR. Therefore, the NC tuned OLSR are more scalable. By reducing the protocol routing load, the routing tables are updated less frequently, calculating **routing paths 27% longer**. Nevertheless, the mitigation of the OLSR related congestion problems by optimized configurations generally allowed to **shorten the packet delivery times**. Besides, these features were obtained while keeping the **degradation of amount of delivered data lower than 5%**.

4.4 Power Aware Proactive Routing for VANETs

The type of routing protocol affects the energy dynamics during the communications in two different ways: first, the routing network load has an influence on the amount of energy used for sending and receiving routing control messages, and second, the generated routing paths affect to which nodes will consume energy in forwarding the packets. In addition, the energy conservation techniques applied in vehicular communications must ensure the QoS requirements of VANET applications (Wu et al., 2010).

After the previous study presented in Toutouh et al. (2012b), which specifically focused on the optimization of the QoS of a proactive routing protocol (OLSR), this section deals with the reduction of the energy consumption of the same kind of protocols since their performance in terms of energy-efficiency is limited due to their need of maintaining the routing tables updated (De Rango et al., 2008).

Several approaches have been proposed to reduce the power consumption of OLSR (De Rango and Fotino, 2009; Sangeeta and Sing, 2011). Additionally, Toutouh and Alba (2011a) presented a study that concluded on the benefits for energy coming from using a QoS optimized version of OLSR. This result motivated the definition of a specific optimization problem to reduce the energy requirements of the VANET communications.

Then, Toutouh and Alba (2012a) analyzed the use of DE algorithm in solving the energy-efficiency OLSR optimization problem for VANETs in an initial study, which resulted that the improved protocol **saved up to 30%** of energy with negligible a negligible reduction of PDR (bellow 9%). After that, Toutouh and Alba (2015b) performed an in depth study in applying NC (PSO, DE, GA, ES, and SA) to reduce the energy requirements of OLSR that succeed in improving the energy **savings up to 33%**, with a significant gains of up to 50% in large dense networks (demonstrating a competitive scalability of this approach).

Although the results provided by these two previous studies are very competitive, the large amount of time required to perform the VANET simulations limited the proposed search methods to work with a reduced population in order to obtain results in reasonable execution times. To overcome this drawback in the previously studied GA (Toutouh and Alba, 2015b), the study presented in the following section proposes to use a parallel evolutionary algorithm (*pEA*) based on GA (*pGA*) for efficiently searching the parameter values of the OLSR protocol.

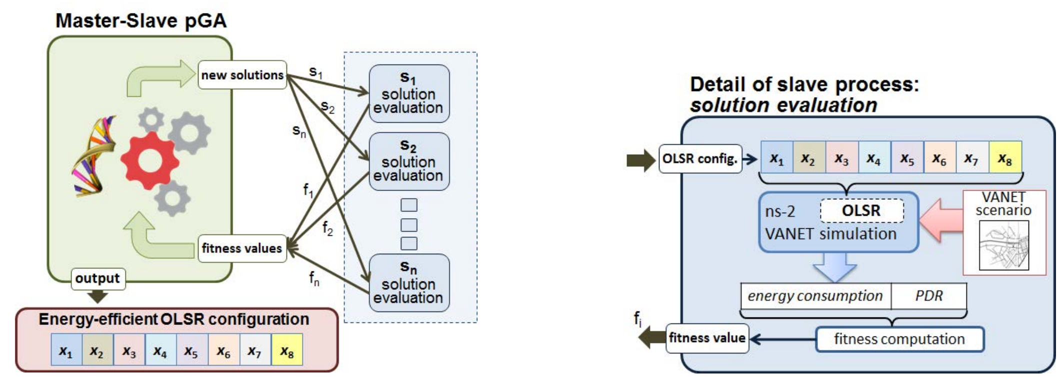
4.4.1 Energy-efficiency OLSR optimization in VANETs

The OLSR routing protocol previously presented in Section 4.3.1 provides high QoS capabilities. However, it suffers from a high energy costs which worsens when increasing the size or density of the network (De Rango et al., 2008). For this reason it has been selected for being fine tuned automatically with NC techniques, with the aim of hopefully reducing the power consumption without incurring a significant lost of QoS.

The automatic search for energy-aware OLSR configurations is carried out by using the energy cost of the communications as the main objective to be optimized. However, since excessive reductions of power consumption of the protocol can cause it to malfunction, PDR is uses as a quality indicator to guarantee a minimum level of QoS in the communications. Thus, the pGA for finding higher energy-efficient parameter values searches the *best* configuration that provides the most energy savings while maintaining PDR within margins of good performance (our target is that degradation in the PDR value is kept below 15% of the PDR achieved with the standard OLSR).

4.4.2 Implementation Details

The pEA proposed here is categorized as a *master-slave* model according the classification by Alba and Tomassini (2002). This model follows a classic functional decomposition of the EA, where different stages of the evolutionary process are performed in different processing elements (CPUs). The evaluation of the fitness function (VANET simulation) is the main candidate to perform in parallel, since it usually requires larger computing time (in order of minutes) than the application of the variation operators. Specifically, in our case, the master process performs the evolutionary search and controls a group of slave processes that evaluate the fitness function (see Figure 4.11).



a) Global master-slave pGA methodology.

b) Details of how one slave process (s_i) performs solution evaluation (f_i).

FIGURE 4.11: Optimization strategy to deal with the power-aware optimization of OLSR.

Multithreading is well suited for multi-core computers, where each thread is executed on a single core. The multithreading master-slave pEA proposed in this study is implemented using the GA skeleton provided by MALLBA and the standard *pthread library*. Additional code is incorporated into the GA skeleton to implement several new features: **i)** to create and manage the pool of threads used for the fitness evaluation; **ii)** to implement the master-slave hierarchy and the communications between master and slaves; and **iii)** to define the synchronization mechanisms between threads. Further details about the implementation are presented in Toutouh et al. (2013).

Problem Encoding

Since the optimization problem is defined over OLSR, the solution is encoded as a vector of eight components in the same manner that the OLSR QoS optimization problem (see Section 4.3.3). The valid ranges for the parameter values are presented in Table 4.5.

Fitness Function

The optimization proposed in this analysis mainly concerns to power-aware communications, so the main component of the fitness function is the energy consumed by the VANET nodes when using a certain OLSR configuration. There is a trade-off between the energy efficiency and the QoS provided by the protocol. Therefore, the fitness function integrates the PDR metric to guide the search to solutions with acceptable QoS. The function in Equation 4.14 is formulated as a minimization problem (minimizing the fitness function).

Equation 4.15 is valid for solutions with a PDR degradation lower than 15% of the reference value. In $fitness_Q(s)$, $E(s)$ and $PDR(s)$ represent the power consumption and the PDR for a given OLSR configuration s, respectively. E_{RFC} and PDR_{RFC} are the reference values for the power consumption and the PDR when using the standard OLSR. Finally, ω_1=0.9 and ω_2=-0.1 are the weights for the energy and PDR contributions, respectively, and Δ=0.1 is a normalizing offset to keep the fitness value in the interval $[0, 1]$ (Toutouh and Alba, 2012a).

A penalization model in Equation 4.16 is applied to keep in the population those solutions with still a lower PDR. The penalized fitness $fitness_P(s)$ takes into account the gap between the PDR of the evaluated solution and the worst PDR value admitted (0.85 $\times PDR_{RFC}$), and the ratio between the energy of the evaluated solution and the reference energy value E_{RFC}.

$$fitness(s) = \begin{cases} fitness_Q(s) & \text{if } PDR(s) \geq 0.85 \times PDR_{RFC} \\ fitness_P(s) & \text{if } PDR(s) < 0.85 \times PDR_{RFC} \end{cases} \tag{4.14}$$

$$fitness_Q(s) = \Delta + \left(\omega_1 \cdot \frac{E(s)}{E_{RFC}} + \omega_2 \cdot \frac{PDR(s)}{PDR_{MAX}}\right) \tag{4.15}$$

$$fitness_P(s) = fitness_Q(s) + \left((0.85 \cdot PDR_{RFC} - PDR(s)) \cdot \frac{E(s)}{E_{RFC}}\right) \tag{4.16}$$

Parallel EA Operators

In the literature, a classic GA has been applied for protocol tuning in a previous studies (García-Nieto et al., 2010; Toutouh et al., 2012b). However, although it offered competitive results, that algorithm suffered from low population diversity and early stagnation. For this reason, in this work the *diagonal uniform initialization*, the classic *arithmetic recombination*, and *OLSR-μ mutation* operators are applied in the GA (see Section 4.1.3).

4.4.3 Experimental Results

This section presents and analyzes the experiments carried out to solve the power-aware routing problem on OLSR. The experimental analysis is performed in a cluster with Opteron 6172 Magni-core processors at 2.1 GHz and with 24 GB RAM.

VANET Instance for Fitness Computations

In this study, the U1 and U2 urban areas are used to define the VANET scenarios for the simulations (fitness computations), both with 20 vehicles moving along the roads during three minutes. The small-sized (U1) is applied in the pGA parameter setting experiments, while the medium-sized (U2) is utilized in the optimization of OLSR parameters using the pGAs. In these scenarios, ten pairs of these vehicles exchange data created by a constant bit rate data generator (CBR) application during one minute. Toutouh et al. (2013) describes further details of the analyzed VANET instances.

Parameter Settings of the pGA

A parameter setting analysis is performed to study the most competitive values for the crossover probability (p_C) and the mutation probability (p_M) in the pGA. The analysis is done over a small VANET defined in scenario U1 (with reference values $E_{RFC} = 5680$ and $PDR_{RFC} = 88.23\%$). The population size of the parallel GA is fixed to 24 individuals (24 threads), and the stopping criterion is set at 100 generations. The candidate values for the parameters are: p_C: 0.5, 0.7, 0.9; and p_M: 0.25, 0.0125, 0.006125. According to the results, the most competitive performance is obtained with the p_C=0.7 and p_M=0.25. The whole parameter settings results are presented in Table B.8 in Appendix B.

Numerical Analysis

By using the parameters found ibefore, now we apply the pGA to the problem. Table 4.11 summarizes the energy-efficiency optimization results. Three pGA variants are studied: implementations using 8, 16, and 24 individuals, and the same number of execution threads (named pGA-8, pGA-16, and pGA-24, respectively). In order to provide a baseline for the comparison, the analysis includes the results obtained with a sequential (single thread) GA (using a population of 8 individuals).

Table 4.11 reports the average, relative standard deviation, and best fitness results obtained in 30 independent executions performed for each algorithm. In addition, the power consumption and PDR values obtained with the best OLSR configuration found, and the gaps with respect to the standard RFC parameterization are also presented.

TABLE 4.11: Experimental results of power-aware OLSR optimization: pGA evaluation.

Algorithm	Fitness			Metrics		GAP RFC	
	Average	Std. dev.	Best	Energy	PDR	Energy	PDR
sequential GA	0.7521	2.66%	0.7025	6909.12	80.48%	24.11%	-6.64%
pGA-8	0.7058	1.88%	0.6730	6551.89	74.74%	28.03%	-12.38%
pGA-16	0.6883	1.69%	0.6621	6446.80	75.20%	29.19%	-11.92%
pGA-24	**0.6774**	1.37%	**0.6482**	6305.58	75.14%	**30.74%**	-11.98%

The pGA-24 obtains the most competitive results (lowest average, deviation, and best fitness values). As expected, the sequential GA performs the worst. The improvements in the fitness values bring forth a significant decrease in the power consumption: more than **30%** of reduction with respect to the standard OLSR configuration is achieved for the best configuration found using pGA-24, while the PDR degradation remained below 12%.

In order to determine the significance of the comparison, a statistical analysis is performed over the results distributions for each parallel GA. The non-parametric Kruskal-Wallis statistical test is performed with a confidence level of 95%, to compare the distributions for pGA-8, pGA-16, and pGA-24 because the distributions of the results are not normally distributed. The small p-values reported (<0.05 in all cases) indicate that the fitness improvements can be considered statistically significant, thus the parallel GA using 24 threads is the best algorithm out of all the studied methods.

Computational Efficiency Analysis

This section discusses the *speedup* (S_m) and the *computational efficiency* (e_m) computed according to the equations 3.15 and 3.16 presented in Chapter 3. Table 4.12 shows the average and best run times, and the values of the speedup and efficiency the pGAs when using 8, 16, and 24 threads. These results suggest that significant reductions in the required run times are obtained when using the parallel implementations with respect to a sequential version.

TABLE 4.12: Performance comparison of the proposed pGAs (computational cost and efficiency).

Algorithm	Execution time (s)		Speedup		Efficiency	
	Average	Best	Average	Best	Average	Best
pGA-8, 8 threads	11113.73	9235.71	5.80	6.86	0.72	0.86
pGA-16, 16 threads	13192.70	12440.05	11.81	12.63	0.74	0.79
pGA-24, 24 threads	20239.02	13670.90	**19.10**	20.12	**0.80**	0.84

The results in Table 4.12 demonstrate that the proposed master-slave model is a useful choice to significantly reduce the execution times of the pGAs. Despite following a synchronous paradigm (that tends to generate idle times due to the synchronization of the execution threads), the parallel GAs show an almost-linear speedup behavior. The average efficiency values obtained are **greater than 70%** for the three implementations studied, and a maximum average of **80%** is achieved when using the pGA-24.

4.4.4 Power Aware OLSR Validation

In order to confirm the efficacy of the results obtained in the experimental analysis, a set of validation experiments are conducted to compare the performance of the best OLSR configurations found using each pGA against the standard OLSR (RFC) configuration. The validation experiments involved simulations performed over 36 different urban VANET scenarios, which comprise the simulation of three different road traffic densities defined in the U2 and the U3 areas and six different types of applications (three *traffic densities* $\times$ two *urban areas* $\times$ six *applications* = 36). The definition of the simulation environment is detailed in Toutouh et al. (2013).

This analysis evaluates the energy consumption in transmitting (E_{send}) and receiving (E_{recv}) modes, as well as the total energy (E_{total}) and total energy per vehicle ($E_{tot\times v}$). Additionally, the non-parametric Friedman statistical test is performed over the energy-efficient results. The PDR, the NRL, the E2ED, and the RPL are also included in the study. The results of the whole validation experimentation are included in tables B.10 and B.11 in Appendix B.

TABLE 4.13: Results of the validation experiments of the power-aware OLSR.

Config.	Friedman	Energy metrics				QoS metrics			
	rank.	E_{sent}	E_{recv}	E_{total}	$E_{tot\times v}$	PDR	E2ED	NRL	hops
pGA-24	**1.92**	**13012.84**	**5928.54**	**18941.37**	**527.51**	59.22%	**269.33**	**3.45%**	1.46
pGA-16	1.94	13383.40	6163.63	19547.03	547.34	60.64%	274.34	3.63%	1.46
pGA- 8	2.94	13390.95	6147.99	19538.93	547.67	58.64%	283.85	3.67%	1.54
RFC	3.94	19572.25	12102.03	31674.29	877.33	**67.89%**	506.26	25.22%	**1.20**

Table 4.13 summarizes the experimental results by showing the Friedman ranking and the average values for each metric (the best values are marked in bold). According to these results, significant reductions are obtained when using the OLSR parameterizations computed by using the three pGAs. The configuration found by the pGA-24 is the most efficient parameterization for OLSR in VANETs (first ranked by Friedman), allowing a reduction of up to **40.2%** in the power consumption. Figure 4.12 illustrates the GAP of energy regarding the dimension of the simulated scenarios.

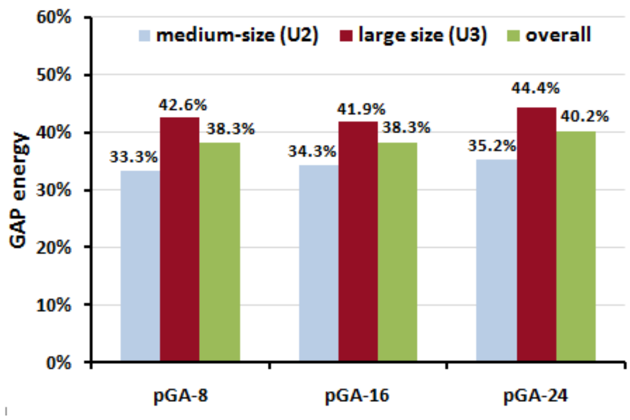


FIGURE 4.12: GAP of energy of the energy-efficient OLSR regarding the scenario dimension.

The results in Figure 4.12 demonstrate that significant improvements in the power consumption are obtained when using the configuration found with pGAs. In addition, the energy reductions with respect to the standard RFC configuration increase for the largest scenarios. The configuration found by pGA-24 achieved up to 44.4% of improvement in average for the largest scenarios. These notable improvements confirm previous claims about the inefficiency of the standard OLSR configuration in large VANET scenarios with high traffic density, already suggested by previous experimental evaluations (De Rango et al., 2008). All the previous results demonstrate the efficacy of the proposed NC methodology to compute accurate energy-aware OLSR configurations.

The energy-efficient OLSR parameterizations, in addition to obtain high power savings, reduce extremely the routing overhead (NRL) and the communication delays (E2ED). These results are obtained without suffering large reductions in the PDR (<8 %) or increments in the length of the routing paths. This is an acceptable value for the loss in the QoS, when taking into account the important advantages achieved.

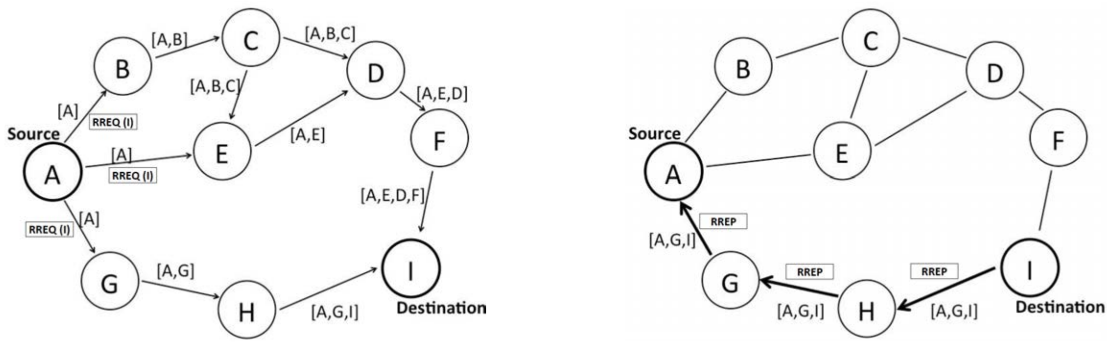
4.4.5 General Discussion on Power-Aware Routing

The proposed improvement in the use of NC techniques to deal with routing optimization in VANET, i.e., the use of parallelism, provided an **almost-linear speedup**, obtaining efficiency values greater than **80%**. This allowed the parallel GAs presented here to outperform the energy-efficiency of the previous power-aware routing approaches using classic NC algorithms. The average energy reductions achieved by pGA-24 are up to **40.2%**, in contrast to the 30% and the 33% obtained in Toutouh and Alba (2011a) and in Toutouh and Alba (2015b), respectively; keeping a higher PDR. Also, significantly better energy savings (up to 77.54%) are computed for large and dense VANET scenarios. In addition, the energy-aware OLSR configuration found in this study significantly reduces the network overload and communication delays. All these important features are obtained while only suffering a bounded PDR degradation (less than 8%), what means a promising result for real applications.

4.5 Efficient QoS for Reactive Data Routing for VANETs

Reactive or on-demand routing protocols have been also analyzed to be used in VANETs (Chauhan and Dahiya, 2012; Ding et al., 2011). The main difference between reactive and proactive protocols is that the first ones determine routing paths only when there is any data to send and the second ones attempt to maintain routes to all destinations at all time. Therefore, reactive routing protocols can be considered as the flip-side of the proactive ones.

In reactive protocols, as AODV, if a node wants to start a communication with another node to which the route is unknown, it initiates a global search procedure to find the destination. This operation is based on classical flooding search algorithms. Indeed, a routing request message (RREQ) is flooded to other nodes. Neighbor nodes which do not know an active route for the requested destination forward the RREQ packet to their neighbors, until an active route is found or the maximum number of hops is reached. When the RREQ reaches the destination or intermediate node with a valid route entry to the destination, a route reply message (RREP) is sent back in a unicast manner to the requester node. Figure 4.13 summarizes this procedure (AODV operation).



a) Route request packet delivery. b) Route reply packet delivery.

FIGURE 4.13: AODV route discovery procedure.

The main advantage of reactive protocols is that they generate less overhead and provide more reliable routing than proactive ones, but at the (long delay) cost of finding the optimal route. In turn, nodes do not utilize periodic messages, with a consequent energy advantage in battery consumption (De Rango et al., 2008).

Specifically, this study analyzes the optimization of the QoS of a reactive protocol (AODV) in VANETs, because, as it happens with the other protocols analyzed before, its performance can be improved by finding parameterizations better adapted to vehicular communications. Here, we focus on the optimization of two important aspects in VANET communications: the reliability and communication delays. This is principally due to the competitive performance of this protocol in terms of overhead generated.

This protocol has been optimized previously in different studies. A preliminary study presented by García-Nieto and Alba (2010) applied several sequential mono-objective NC techniques to the optimization AODV obtaining promising results. In turn, a parallel PSO (*pPSO*) was also analyzed in dealing with this problem, outperforming the classic PSO (Toutouh and Alba, 2012c). Recently, Said and Nakamura (2014) proposed an asynchronous pEA to tackle the optimization of the same protocol.

All these previous studies defined the AODV QoS optimization problem as a mono-objective optimization problem, therefore, they obtained a single solution. In contrast, here, the problem is defined as a multi-objective optimization one, thus obtaining a set of accurate solutions that offer different trade-offs between the objectives.

4.5.1 AODV Routing Protocol for VANETs

AODV (Perkins et al., 2003) is a reactive distance vector (on-demand) routing protocol for mobile ad hoc networks designed to overcome the overhead problem of its precedent, DSDV. As a reactive protocol, AODV determines the routes when a source node has data traffic to send, and it maintains just the paths that are currently in use. Thus, it reduces the routing overload generated by proactive routing protocols to maintain the routing paths at any time. The operation shown in Figure 4.13 and the performance of AODV are significantly influenced by the value of its 11 main control parameters that can be grouped in: (1) five timeout timers: *HELLO_INTERVAL*, *ACTIVE_ROUTE_TIMEOUT*, *MY_ROUTE_TIMEOUT*, *NODE_TRAVERSAL_TIME*, and *MAX_RREQ_TIMEOUT*; (2) three decision variables used in the process of updating and maintaining the routing tables: *NET_DIAMETER*, *ALLOWED_HELLO_LOSS*, and *REQ_RETRIES*; and (3) three counters and decision variables that control the process of discovering new routing paths: *TTL_START*, *TTL _INCREMENT*, and *TTL_THRES-HOLD*. The AODV RFC 3561 suggests a generic MANET parameterization (see Table 4.14) that has been also used in vehicular networks (Chauhan and Dahiya, 2012).

TABLE 4.14: Main AODV parameters and RFC 3561 specified values.

Parameter	Type	Range	Standard configuration
HELLO_INTERVAL	$\mathbb{R}$	[1.0, 20.0]	1.0 s
ACTIVE_ROUTE_TIMEOUT	$\mathbb{R}$	[1.0, 20.0]	3.0 s
MY_ROUTE_TIMEOUT	$\mathbb{R}$	[1.0, 40.0]	6.0 s
NODE_TRAVERSAL_TIME	$\mathbb{R}$	[0.01, 15.0]	0.040 s
MAX_RREQ_TIMEOUT	$\mathbb{R}$	[1.0, 100.0]	10.0 s
NET_DIAMETER	$\mathbb{Z}$	[3, 100]	35
ALLOWED_HELLO_LOSS	$\mathbb{Z}$	[0, 20]	2
REQ_RETRIES	$\mathbb{Z}$	[0, 20]	2
TTL_START	$\mathbb{Z}$	[1, 40]	1
TTL_INCREMENT	$\mathbb{Z}$	[1, 20]	2
TTL_THRESHOLD	$\mathbb{Z}$	[1, 60]	7

4.5.2 Multi-objective AODV Optimization in VANETs

AODV provides moderate QoS when it is used in vehicular communications, although the low routing load generated by reactive protocols makes it scalable to be used in large VANETs.

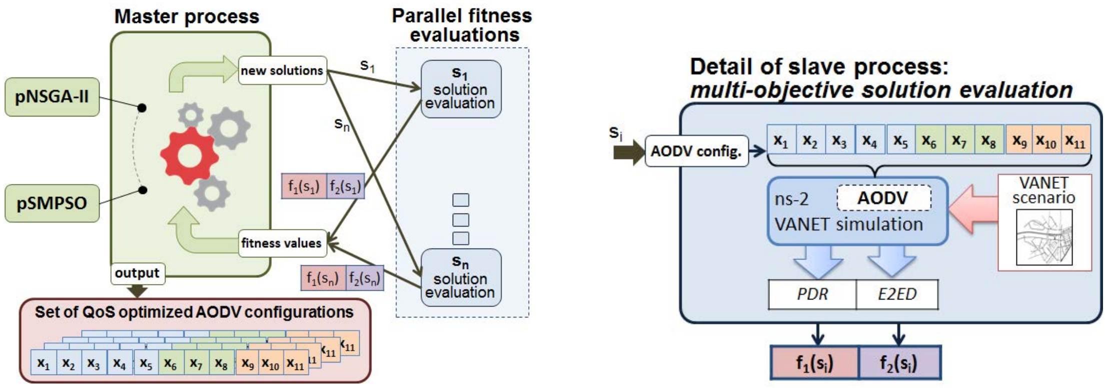
This analysis is aimed at improving the reliability, evaluated in terms of PDR, and the communication delays (E2ED). Therefore, the main idea is to find configuration parameters that maximize the PDR and minimize the E2ED. However, E2ED increases critically with PDR. This is principally because the possibility of collisions increases with the number of the packets traveling through the network: the nodes take longer to relay/send the packets. The opposite occurs when decreasing PDR. Thus, a multi-objective (MO) optimization problem is defined with the aim of discovering a set of efficient AODV parameters based on their performance in terms of PDR and E2ED. This problem is named AODV multi-objective QoS optimization (AODV MO-QoS).

4.5.3 Implementation Details

The AODV MO-QoS problem is treated by using two different multi-objective algorithms (MOAs): a multi-objective evolutionary algorithm, the *Non-dominated Sorting Genetic Algorithm-II* (NSGA-II) (Deb et al., 2002), and a multi-objective swarm optimization method, the *Speed-constrained Multi-objective PSO* (SMPSO) (Nebro et al., 2009). The competitive computational efficiency of the parallel NC algorithms in solving off-line optimization of protocols in vehicular communications (Toutouh and Alba, 2012c; Toutouh et al., 2013; Said and Nakamura, 2014) motivated the use of parallel implementations of the two utilized MOAs (pMOAs).

The application of pMOAs mitigates the main issues of most precious work in optimizing VANET routing (Patil and Dhage, 2013; Toutouh et al., 2012b; Zukarnain et al., 2014), which are: i) the use of single-objective methods to optimize an aggregated objective function, obtaining a single biased solution, and ii) the relatively low number of fitness evaluations carried out during the search process due to the high computational costs of the VANET simulations, needed to perform such an operation.

Figure 4.14.a summarizes the operation of the two analyzed pMOAs, in which the master process performs most of operations of NSGA-II or SMPSO and the n slave processing units carry out the solution evaluation. The slave process is showed in Figure 4.14.b: it receives a given solution s (AODV configuration), which is simulated, and it returns the evaluation of the objective functions regarding to PDR and E2ED.



a) Global master-slave methodology.

b) Details of how one slave process performs solution evaluation of f_1 and f_2 objective functions.

FIGURE 4.14: Methodology applied to solve the MO-AODV problem.

Problem Encoding

As AODV is governed by 11 configuration parameters, the solution is encoded as a vector of 11 components. The valid ranges for each one of the parameter values are presented in Table 4.14. Figure 4.15 illustrates a representation of of this vector.

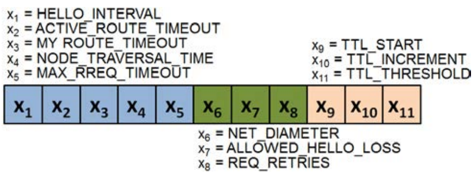


FIGURE 4.15: Solution encoding for the QoS AODV tuning problem.

Multi-objective Solution Evaluation

This study focuses on maximizing the PDR and minimizing the E2ED of AODV in VANETs. Thus, a multi-objective optimization problem (MOP) is defined (the multi-objective AODV MO-QoS problem) in which there are two objectives to be optimized, that are defined by two *fitness functions* ($f_1(s)$ and $f_2(s)$).

The fitness functions are evaluated after performing a VANET simulation configuring AODV with a given s parameterization. The $f_1(s)$ function is given by the expression in Equation 4.17, in which $PDR(s)$ is the average PDR achieved by all the VANET nodes. As $PDR(s)$ is a value from 0 to 100 (100 is the value achieved when all the data packets are delivered), then $f_1(s) \in [0, 100]$. The idea is that the problem of maximizing PDR has been changed to a problem of minimizing $f_1(s)$ to ease the representation. The $f_2(s)$ function evaluates the $E2ED(s)$ which is the average delivery time of all the data packets (see Equation 4.18). Thus, $f_2(s)$ has to also be minimized. The E2ED time is given in milliseconds (ms). Therefore, the AODV MO QoS optimization problem is given by minimizing $f_1(s)$ and $f_2(s)$.

$$f_1(s) = 100 - PDR(s) \quad (4.17) \qquad\qquad f_2(s) = E2ED(s) \quad (4.18)$$

Parallel Multi-objective Operators

The two NC algorithms utilize the *diagonal uniform initialization*. The diversity is introduced by applying the *arithmetic recombination* and the *AODV-μ mutation* operators, in the case of the parallel NSGA-II (*pNSGA-II*), and the *AODV-μ mutation* operator, in the case of parallel SMPSO (*pSMPSO*). The operators are defined in Section 4.1.3.

4.5.4 Experimental Results

The experiments carried out to solve the AODV MO-QoS optimization problem are presented in this section. The pMOAs are implemented using jMetalCpp framework (Durillo and Nebro, 2011) and the standard *pthread library*. The experimental analysis is done in a cluster with Opteron 6172 Magni-Core at 2.1 GHz with 24 GB RAM.

VANET Instance for Fitness Computations

In this study, the U2 urban VANET scenario with 30 vehicles moving along the roads during three minutes is used for fitness functions computations (ns-2 simulations). In this scenario there are 15 nodes transmitting data at 256 kilobits per second (*kbps*) during one minute. Toutouh and Alba (2015c) details this VANET instance.

Parameter Settings of the Parallel NC Algorithms

A parameter settings initial analysis is performed to set the crossover (p_C) and mutation (p_M) probabilities of the pNSGA-II and the mutation probability (p_M) of pSMPSO. The experiments are carried out on both pMOAs with a population/swarm size of 24 solutions (24 threads) stopping after 300 generations. The candidate values for the parameters are for p_C: {0.3, 0.5, 0.7, 0.9}; and for p_M: $\{\frac{1}{4L}, \frac{1}{2L}, \frac{1}{L}, \frac{1}{0.5L}\}$ being L the number of components of the solution vector ($L = 11$). Each configuration of each algorithm is independently executed ten times and the *hypervolume* I_{HV} metric (Deb, 2001) is compared among different configurations of the same algorithm. Table B.12 in Appendix B shows the results for the parameterizations of the algorithms analyzed here. The best results are obtained configuring pNSGA-II by using p_C=0.9, $p_M=\frac{1}{4L}$=0.023, while pSMPSO with $p_M=\frac{1}{L}$=0.091.

Defining an Empirical Stop Criterion

In order to compare both analyzed algorithms the stop criterion is set as obtaining a Pareto front with a given quality in terms of the hypervolume value. As the AODV MO-QoS is a new and open problem, no optimal Pareto front is known. Therefore, some initial experimentation is performed to compute an approximated Pareto front to later evaluate the hypervolume during the execution of the algorithms when solving the problem. These initial runs stop after computing 450 generations. The union set of all non-dominated solutions computed for both algorithms is considered as the *optimized Pareto front*. The median hypervolume computed by all the runs is 0.785. Thus, the stop criterion applied is, first, achieving a hypervolume value equal or higher than 0.785, or second, performing a given maximum number of generations.

Numerical Analysis

This section compares the performance of the two NC algorithms studied on the optimization of the QoS of AODV in VANETs after performing 30 independent runs of each method.

The comparison of the Pareto fronts approximations are carried out in terms of *epsilon* (I_ϵ) and *spread* (I_Δ) values (Deb, 2001). The hypervolume is not used because it is used as stop criterion and there is not a significant difference between the two pMOAs in this metric (see Figure 4.16). In order to determine the significance of the comparisons, Wilcoxon statistical test with a confidence level of 99% (*p-value*<0.01) is applied to compare each metric because they are not normally distributed.

Table 4.15 shows the minimum (Min), median (Med), and maximum (Max) values obtained for each metric and algorithm. Please, note that the minimum hypervolume values obtained by each algorithm (0.772 by pNSGA-II and 0.777 by pSMPSO) are lower than 0.785 (the threshold value used as the stop criterion). This occurs because four independent runs have not achieved such hypervolume value and they stopped when they performed the maximum number of generations (450).

TABLE 4.15: AODV MO-QoS experimental results.

Metric	pNSGA-II			pSMPSO		
	Min	Med	Max	Min	Med	Max
I_{HV}	0.772	**0.786**	0.791	0.777	**0.786**	0.799
I_ϵ	**1.741**	2.634	3.949	2.195	2.807	4.493
I_Δ	0.511	**0.619**	0.953	0.603	0.706	0.911

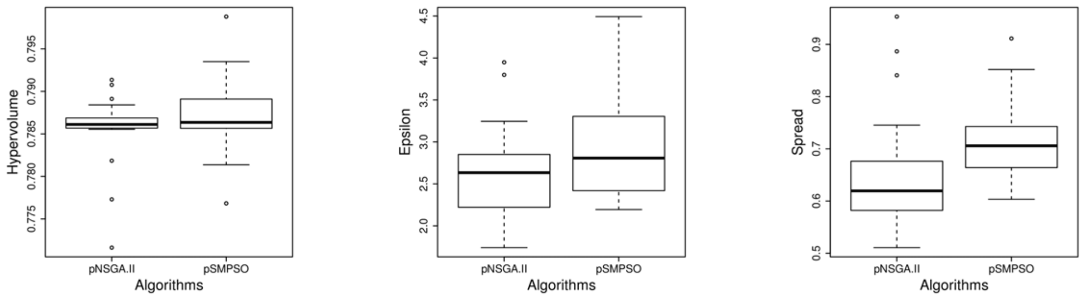
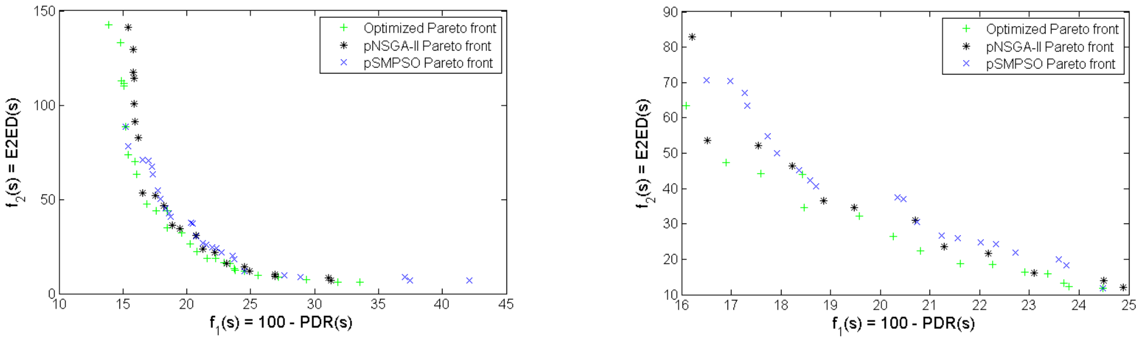


FIGURE 4.16: Quality metrics of the Pareto fronts computed in solving AODV MO-QoS.

The epsilon values of pNSGA-II are in general better (lower) than the ones computed by pSMPSO (see Figure 4.16 and Table 4.15). Therefore the convergence of pNSGA-II is more competitive than the one of pSMPSO. The spread values achieved by the pNSGA-II computed Pareto fronts are also better (lower) than the ones obtained by pSMPSO. Thus, the diversity of the pNSGA-II fronts is better than the ones computed by pSMPSO. These results are confirmed by the Wilcoxon test results because the *p-values* computed are 0.0036 and 0.0024 for epsilon and spread, respectively.

Figure 4.17 illustrates two Pareto fronts obtained by each pMOA and the *optimized Pareto front* (the complete fronts in the left and a zoom of a given region of the shown fronts in the right). The selected Pareto fronts are the ones that obtain the median hypervolume value for each algorithm. The pNSGA-II solutions are better distributed among the optimized Pareto front (see Figure 4.17.a). The pSMPSO front does not contain solutions that best minimize $f_1(s)$ (maximize PDR), while it has solutions that sharply reduce the E2ED times and critically worsen the PDR. Figure 4.17.b shows that most of pNSGA-II solutions dominate the ones computed by pSMPSO. This confirms the aforementioned results that concluded that pNSGA-II presents the best performance when the hypervolume is set as the stop criterion for solving AODV MO-QoS problem.



a) Complete Pareto fronts of median run. b) Detail of Pareto fronts of median run.

FIGURE 4.17: Pareto fronts computed when solving AODV MO-QoS.

Computational Efficiency Analysis

This section evaluates the performance of parallel algorithms in terms of speedup and computational efficiency according to equations 3.15 and 3.16 presented in Chapter 3 (see Table 4.16). Both algorithms, NSGA-II and SMPSO (with the same configurations as the pMOAs) are executed sequentially to compute $E[T_1]$ (the run time of the one-thread version). As the average execution times of these sequential versions of NSGA-II and SMPSO are 94.5 hours (3.9 days) and 173.1 hours (7.2 days), respectively, we were bound to perform just 10 independent runs of each algorithm due to the limited access to the computational platform (and the large number of tests done for the rest of this research work).

TABLE 4.16: Performance comparison of the proposed pMOAs in solving AODV MO-QoS.

Algorithm	$E[T_{24}]$	$E[T_1]$	Speedup	Efficiency
pNSGA-II	311.737	5668.040	**21.614**	**0.901**
pSMPSO	422.190	10383.449	20.829	0.868

As it happened in the analysis performed in Section 4.4, in the proposed pMOAs, the evaluation of the fitness function is the most consuming part within the algorithm, since the ns-2 simulations demand large computation costs. The results in Table 4.16 demonstrate that the proposed master-slave model is a successful choice to significantly improve the efficiency of the multi-objective metaheuristic algorithms analyzed in this study. The speedup values are larger than 20.8, obtaining highly satisfactory efficiency values both for pNSGA-II (90.1%) and pSMPSO (86.8%).

4.5.5 Improved AODV Validation

A set of validation experiments are carried out to confirm the real applicability of our proposal. Thus, a *representative solution* (AODV configuration) obtained by each pMOA is compared against other state-of-the-art ones, which are: the standard AODV RFC 3651 (*RFC*), the one proposed in García-Nieto and Alba (2010) found applying a PSO (*GN*), and the one tuned using a pPSO (*pPSO*) proposed in Toutouh and Alba (2012c).

The solutions selected as the representative one of each pMOA are the ones that minimize the distance to the *ideal vector* (Coello et al., 2007). Table B.13 in Appendix B presents these configurations. However, one of the most important advantages of using multi-objective NC techniques is that they return a set of accurate solutions (configurations), which offer different trade-offs between the objectives. Thus, these configurations may be selected according to the actual conditions of the VANET.

The comparison is carried out taking into account 30 VANET scenarios defined in the three urban areas defined in Section 4.1.1. These scenarios comprise ten urban VANET traffic conditions and three different applications. Toutouh and Alba (2015c) details these urban VANET scenarios. Table 4.17 summarizes the results by showing the median PDR and E2ED for the whole experimentation (Table B.14 presents the whole experimentation results). The best results are marked in bold. Moreover, Friedman Rank statistical test is applied because the results are not normally distributed. The confidence level is set to 99% (*p-value*=0.01). The statistical test results are presented in Table 4.18.

TABLE 4.17: Median values of the whole AODV validation experiments.

configurations	PDR (%)	E2ED (ms)
RFC	64.510	59.787
pPSO	59.269	17.183
GN	59.787	**10.753**
pNSGA-II	67.015	97.206
pSMPSO	**69.203**	65.681

TABLE 4.18: Friedman Rank test results of the AODV validation experiments.

PDR		E2ED	
Configs.	Rank	Configs.	Rank
pSMPSO	**3.93**	**GN**	**1.53**
pNSGA-II	3.50	pPSO	1.87
RFC	3.20	RFC	3.40
pPSO	2.47	pSMPSO	3.60
GN	1.90	pNSGA-II	4.60

Concerning the PDR results, the best median values are: first pSMPSO, second pNSGA-II, and third RFC. The results of the statistical test in Table 4.18 confirm that these three configurations are the best ranked ones and in the same order. Regarding the times required to delivery the packets (E2ED), the configurations obtained by the single-objective algorithms (GN and pPSO) obtain the best results for the analyzed scenarios. The Friedman Rank statistical test confirms these results, because the test ranked GN as the first and pPSO as the second best configurations. Comparing just the configurations obtained by the pMOAs, pSMPSO performed better than pNSGA-II. In short, the selected solution to represent pMOAs in the validation experiments provided the best PDR results, while they suffered from slightly longer E2ED. Moreover, taking into account just pMOAs solutions, the pSMPSO solution performed the best.

4.5.6 General Discussion on Efficient MO-QoS Optimization

This last research work has been motivated by the need of a deep experimentation in off-line optimization in VANETs with NC algorithms. In this sense, the problem has been formulated as a multi-objective optimization problem with the aim of maximizing the PDR and minimizing the E2ED. The problem has been addressed by using two pMOAs that perform the search with high competitive results from the point of view of parallelism (computational efficiency is 86.8% for pSMPSO and 90.1% for pNSGA-II).

The AODV configurations computed by the pMOAs improve the PDR obtained by the other tstate-of-the-art AODV parameterizations (the standard AODV RFC and two improved ones by using other NC algorithms), while not leading to a degradation of the other network performance metrics. However, other solutions (AODV configurations) in the optimized Pareto front may offer different trade-off of QoS metrics.

Analyzing the optimization process, pNSGA-II significantly outperformed pSMPSO in terms of diversity (spread) and convergence (epsilon) in solving AODV MO-QoS optimization problem. In addition, pNSGA-II required lower computation costs: fewer generations and shorter run times.

4.6 Conclusions

This chapter has analyzed the application of NC algorithms for the off-line optimization of VANET software communication protocols. The works carried out in this research are a key part of the thesis presented here. For this reason, we have presented the design of algorithms, their implementations, initial results (with previous parameterization studies), validation other scenarios and extended comparisons. The off-line optimization requires the joint implementation of a NC algorithm and a realistic VANET simulation. Thus, we have included experimentation and analysis in order to improve the state-of-the-art approaches in both aspects: defining competitive search algorithms and providing new and realistic VANET scenarios for accurate simulation.

We have focused on the optimization of a file transfer (VDTP) and two routing protocols (OLSR and AODV). However, nothing prevents the use of the techniques presented here to optimize other kinds of VANET protocols. The parameter configurations tuned by using NC better fit vehicular environments, and therefore, they improve the state-of-the-art ones in terms of QoS and energy-efficiency.

Analyzing the VDTP QoS optimization by applying PSO, DE, GA, ES, and SA; we conclude that the communication configurations offered by PSO increased the effective data rate of the human expert configuration **from 243 KB/s to 300 KB/**s and **from 31 KB/s to 42 KB/s** in urban and highway roads, respectively. In addition, the NC algorithms applied demonstrated a high scalability in dealing with the problem.

Regarding the optimization of the QoS of OLSR by using a set of sequential NC algorithms, the computed accurate protocol configurations outperformed other state-of-the-art ones. They showed **high PDRs (>84%)**, they generated **much smaller routing overhead** than the standard OLSR, and they **reduced the delivery times**. Concerning the optimization process, SA outperformed the other analyzed algorithms, but PSO presented the best trade-off between performance and computation cost.

The same protocol (OLSR) has been optimized in terms of energy-efficiency. For this problem, a pEA was utilized to mitigate the problem of the high computational costs of the fitness evaluations (VANET simulations). This approach provided a **computational efficiency greater than 80%**. The energy-efficient configurations achieved **energy consumption reductions up to 40%**, improving the results of other NC based search procedures presented in previous studies.

AODV has been also optimized in terms of QoS (PDR and E2ED) by formulating a multi-objective optimization problem and addressing it by utilizing NSGA-II and SMPSO. The algorithms were implemented by following the parallel master-slave model, motivated by the high computational efficiency shown in previous studies. In effect, these algorithms provided a **computational efficiency of 87% in pSMPSO and 90% in pNSGA-II**. The computed protocol configurations outperformed other state-of-the-art ones in terms of PDR while keeping the E2ED in the threshold of proper operation.

Finally, there is another study published in off-line optimization that has not been detailed in this chapter because of length constraints. The energy-efficiency optimization of AODV presented in Toutouh et al. (2012a). In this case, the NC algorithm analyzed (DE) to address the problem performs the evaluation of a given solution by using parallel Monte-Carlo simulations of variations over the same VANET scenario, and thus, leading the algorithm to improve the accuracy in the solution evaluation. Using the computed parameterization an average **reduction of 32% in the power consumption** was obtained.

All the results presented in this chapter lead us to confirm the working hypothesis of this thesis: that coupling NC techniques and accurate VANET simulations represents a valid methodology to optimize (to improve), and thus, make practical existing theoretical protocols for vehicular communications.

Chapter 5

On-line Broadcasting Optimization in Vehicular Networks

BROADCASTING beacons in VANETs is crucial for most of VANET safety applications (CVS). In this chapter, we introduce the problem of fair (balanced) beacon broadcasting in VANETs. Then, we propose the *FREEDY* family of distributed dynamic broadcasting algorithms to address the efficient congestion control. The FREEDY algorithms apply their optimization strategy during the communication process. This is different to the other studies presented in the previous chapter that apply the optimization algorithm (e.g., PSO, GA, etc.) previous to the VANET deployment. Finally, we perform a set of experiments over highway VANET scenarios to evaluate these algorithms.

5.1 Introduction

Cooperative Vehicle Safety (CVS) applications presented in Section 2.3 are a very important set of applications provided by vehicular communication systems (Sengupta et al., 2007). CVS applications principally rely on broadcasting short messages (beacons) on the neighborhood (1-hop) defined by the communication range of the nodes (r). Beacons include vehicle kinematics (e.g., position, speed, and acceleration) and other relevant information for several applications or services. VANET nodes are continuously broadcasting beacons (*beaconing*) with a given frequency (*beacon frequency* or *beacon rate*). Figure 5.1 shows an example of a vehicle broadcasting beacons.

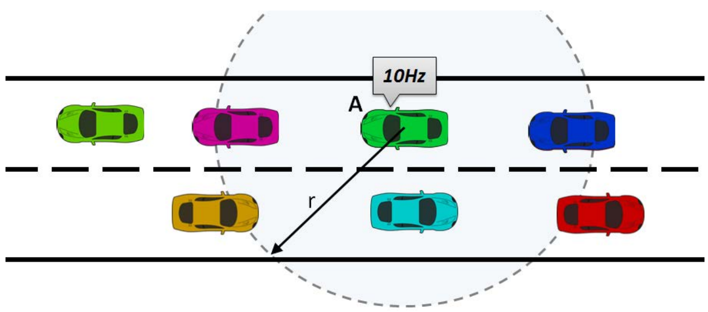


FIGURE 5.1: *Vehicle A* is carrying out CVS communication.

The reliability of CVS, even lead to save lives, is highly dependent on two QoS metrics: the packet loss and the communication delays. As discussed in Section 2.6, a challenging issue in the deployment of VANETs is the network congestion, that aggravates as the scale of the system grows. This is mainly because the critical increase of the periodic beacons that cannot be finally transmitted through the limited channel resources. Network congestion increases the loss of packets and the communication delays (i.e., it degrades the performance and the QoS of the VANETs). This may lead to excessive information inaccuracy and to an eventually failure of CVS (Fallah et al., 2010).

Several strategies have been proposed to address congestion problems in vehicular communications, keeping the communication capabilities of the nodes over a given QoS threshold. Most of them can be included in the following basic schemes (Sattari et al., 2012): **i)** adapting the transmission range of used communication channels, **ii)** adjusting the data rate generation of applications and services, **iii)** hybrid methods by combining the two previous schemes, and **iv)** scheduling data packets in various channels based on their constraints, resources, etc.

This chapter proposes a distributed dynamic greedy broadcasting scheme, that applies congestion control by adjusting beacon rate at each node in order to fit the application requirements and the actual network status (we will call them *FREEDY family*). In the current literature there are some broadcasting algorithms that foster similar strategies, e.g., the *cooperative active safety system (CASS)* presented in Rezaei et al. (2007) and the *situation-adaptive beaconing* introduced by Schmidt et al. (2010).

A greedy algorithm can be seen as an iterative procedure (heuristic) that selects a local optimum at each step while it is solving the problem. In fact, greedy algorithms have been utilized to address optimization problems in many domains such as bioinformatics (Li et al., 2012), security (Liang et al., 2015), and telecommunications (Tan and Kermarrec, 2012).

The present chapter is organized as follows: Section 5.2 formalizes the *fair beacon rate* (FBR) optimization problem. Section 5.3 defines the common framework utilized by the dynamic greedy congestion control methods devised here. Section 5.4 and Section 5.5 detail the Self FREEDY and the Swarm FREEDY algorithms. Section 5.6 presents several experiments and discusses the results. Finally, the conclusion of this analysis are described in Section 5.7.

Let us start with an example to understand the problem and the proposed solution, just to later discuss on beacon frequency and different ways in which we can control and balance the communication channel.

5.1.1 Problem Example

Congestion control mechanisms may produce *unfairness* situations and unbalanced shared medium distribution (nodes with data to transmit suffer from starvation, while other nodes monopolize the medium). In IEEE 802.11 communications, mechanisms based on RTS/CTS protocol have been defined to mitigate *unfairness* when Carrier-Sense Multiple Access (CSMA) is used (Tanenbaum, 2002). However, these solutions cannot be directly applied in vehicular broadcasting for CVS application (Wischhof and Rohling, 2005). One strategy to address congestion in VANETs is utilizing *fair* beaconing. *Fairness* in VANET communications can be seen as the situation in which the nodes located close to each other are able to broadcast beacons with similar (*balanced*) beacon rates while avoiding network congestion.

Figure 5.2 illustrates a very simple CVS example about the difference of using fair beaconing or not. In this example, it is assumed that the beacon rates can be adapted from 1 to 10 beacons per second (*Hertz*, Hz), the maximum channel occupancy in terms of beacons per unit time also called maximum capacity of the channel ($MaxQ$) is 30 beacons per second, the threshold limit ratio over $MaxQ$ that can be used by CVS avoiding system malfunction (α) is 80% (α=0.8), and the transmission (R_{TX}) and the carrier sense (R_{CS}) have the same range (they are marked by dotted circles). According to $MaxQ$ and α values, the maximum number of beacons per second that can be exchanged through the channel is 24 ($\alpha \times MaxQ$=0.8×30=24).

Figure 5.2 shows two separated groups of cars closely located that represent different broadcasting schemes: *Pure CSMA* and *Fair beacon rate*. The first one demonstrates the unfair situation resulting from applying a purely based CSMA based method, which is illustrated by the starvation of cars 3 and 4 that are allowed to transmit just 2 beacons per second. The main reason is that these two nodes need to compete with the others in their carrier sense (cars 1 and 2), which are not aware of that the other nodes need to broadcast data and they transmit beacons at the maximum beacon rate 10 Hz. In the *Fair beacon rate* situation, all the nodes in the same carrier sense (cars 5, 6, 7, and 8) apply a given mechanism to allow all the nodes to transmit the beacons at the same rate (6 Hz) without incurring to a congestion situation.

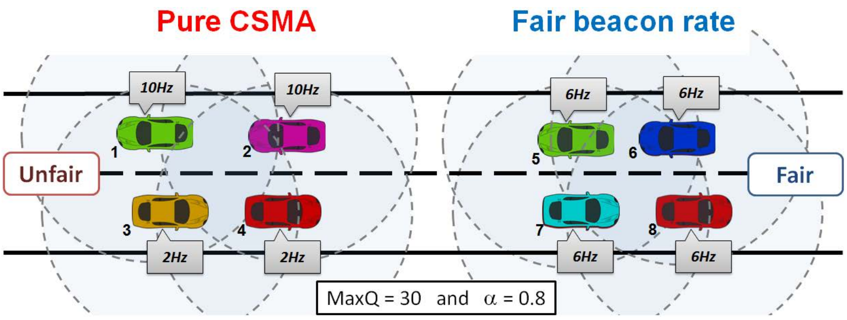


FIGURE 5.2: Simple CVS VANET scenario (CSMA vs. fair beacon rate).

5.1.2 Beacon Frequency as a QoS Metric

Beacons are sent in a broadcast fashion to the neighbor nodes (1-hop) at regular interval times and they are used to make vehicles aware of their environment. They are important for CVS because they contain kinematic information of the vehicle and control data of the software used. Therefore, VANET nodes broadcast beacons to achieve principally two goals: **i)** maintaining constantly aware of the road traffic relevant events in their surroundings to prevent unsafe situations and **ii)** controlling the communications and the applications used in VANETs.

Reliability of CVS applications is really dependent on the capability of the nodes in a local neighborhood to broadcast their beacons. These applications manage more accurate information if the nodes are able to exchange information with higher resolution (beacon rate). Thus, the frequency at which a node can generate beacons and transmit them can be used as a QoS metric since the higher frequency (without generating congestion or node starvation) the more accurate received information (Fallah et al., 2010; Mir et al., 2015).

Due to the channel capacity limitations, it is crucial that nodes broadcast beacons with a suitable beacon frequency according to the current network status. On the one hand, it is large accepted that a high beacon rate can easily result in channel congestion in regions of high vehicle densities, therefore causing a high reduction of beacon delivery and a critical increment communication delays (throughput degradation) (Fallah et al., 2010). On the other hand, larger interval times between consecutive beacons (lower beacon rate) reduce the network throughput and increase the uncertainty of the CVS applications, i.e., nodes might not know the required information about their neighbors for a certain time. Figure 5.3 summarizes how beaconing (channel occupancy) affects the throughput of the network. Thus, beacon rate can be used as a QoS metric that reflects the reliability of CVS applications and the throughput of the VANET.

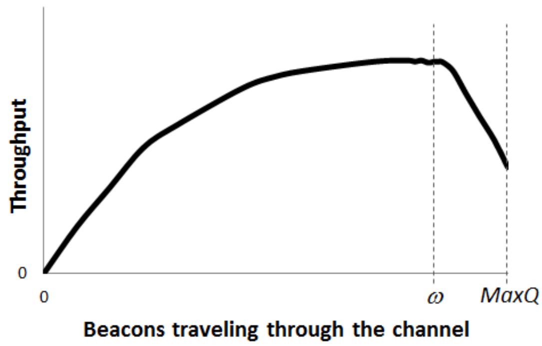


FIGURE 5.3: Representation of the VANET throughput according to the channel occupancy.

5.1.3 Use Case of FBR Utilization in VANETs

In this this study, we define *cluster of nodes* as the set of VANET nodes in which all the nodes are (at least) covered by one of the other nodes of the cluster. In this sense, fairness in beacon rate for VANETs considers three main goals: **i)** maintaining the VANET load under a given threshold to avoid network congestion, **ii)** avoiding the starvation of nodes that have beacons to broadcast, **iii)** and balancing beacon rates (allowing the nodes in a given area exchange beacons with similar rates).

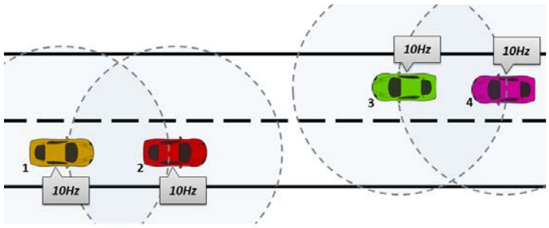
Figure 5.4 illustrates a simple example to show the behavior of VANET nodes when FBR is applied. The main features of this VANET are the same as the ones of the example presented in Section 5.1.1: the beacon rates can be adapted from 1 to 10 Hz (10 Hz = $MaxBR_v$), $MaxQ$=30 beacons per second, and α=0.8, i.e., the maximum number of beacons per second that can be transmitted through the channel to keep the performance is ω ($\omega = \alpha \times MaxQ = 0.8 \times 30 = 24$).

In the example showed in Figure 5.4 there are two cluster of cars: *Group 1* composed by the cars 1 and 2, that travel from left to the right, and *Group 2* composed by the cars 3 and 4, that travel in the opposite direction. At the beginning (see Figure 5.4.a), the two groups of nodes define two different clusters. All the nodes can broadcast beacons at their maximum frequency (10 Hz) because the sum of the beacon rates of the nodes in the same communication range (10+10=20 Hz) do not exceed the maximum beacon rate of the channel (24 Hz). After a given time (see Figure 5.4.b), both groups of cars define a single cluster. They now have to adapt their beacon rate (FBR) in order to avoid network congestion and starvation of the nodes because, in this case, the sum of their beacon rates (10+10+10+10=40 Hz) exceeds 24 Hz. Therefore, they adapt their beacon frequency and change it from 10 to 6 Hz to maintain the load of the channel under the defined threshold (24 Hz). Finally, the cars break up the cluster and build two different ones as the first case (see Figure 5.4.c). Thus, the network status are the similar and the nodes can broadcast beacons again at the maximum frequency.

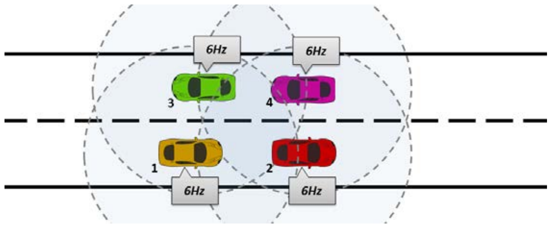
As it can be see in the example, the application of FBR in VANETs requires the dynamic adaptation of the beacon frequency taking into account the current status of the network. Thus, the nodes need to have the information about the actual network load to be able to compute the current FBR.

5.2 Fair Beacon Rate Optimization Problem

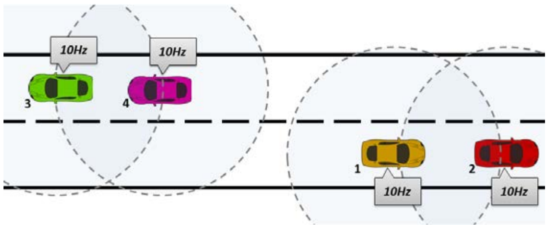
This section formally defines the optimization problem of computing the beacon rates that allow fair beacon VANET communications for CVS, the *fair beacon rate (FBR) optimization problem*. The information required to adapt the beacon rates to the network status is the current



a) *Time 1*. No beacon rate adaptation is needed ($\sum_{v}^{v\in cluster} MaxBR_v < \alpha \times MaxQ$).



b) *Time 2*. Beacon rates are adapted. ($\sum_{v}^{v\in cluster} MaxBR_v > \alpha \times MaxQ$).



c) *Time 3*. No beacon rate adaptation is needed ($\sum_{v}^{v\in cluster} MaxBR_v < \alpha \times MaxQ$).

FIGURE 5.4: VANET example in which nodes apply FBR.

network load (channel occupancy). The channel occupancy is measured by the number of beacons that travel through the channel per time unit. The analysis of the channel occupancy in VANETs can be carried out by monitoring the length of the queues (in the MAC sublayer of IEEE 802.11p) in a given window of time (Han et al., 2012), because the queues contain the beacons received that have been broadcasted by all the neighbor nodes (Sattari et al., 2012). Therefore, in this study, we use the length of the queues to determine the channel occupancy.

The FBR computation problem considers:

- The *maximum allowed channel occupancy* ($MaxQ$). $MaxQ$ in practice represents the maximum value of queues length, i.e., the number of beacons that can be in the queue without representing network overload (congestion).

- A *threshold limit ratio* over the maximum channel occupancy $\alpha \in [0, 1]$. If the queue lengths exceed a given *effective capacity of the channel* ω ($\omega = \alpha \times MaxQ$), the protocol considers that the current network load could lead to a congestion situation (see Figure 5.3) which in turns provoke unpredictability in the network performance (Fallah et al., 2010), and therefore, the beacons may reach the destination nodes or not.

- A *set of allowed beacon rate values* BR=$\{br^1, br^2, ..., br^k\}$ that contains all the possible beacon rate values (br^i) that can be selected by the all nodes according to the VANET application QoS requirements.

- Given a vehicle v that belongs the VANET, the $NN(v)$ function returns the set that contains all the nodes inside its network coverage (1-hop neighbor nodes). This value is computed taking into account all the different source nodes of the received beacons.

The problem consists in finding $br_v \in BR$ for each node v to optimize two objectives:

1. Maximizing number of the beacons traveling through the shared medium in terms of the ratio of the channel occupancy ($\eta(v)$ in Equation 5.1).
2. Minimizing the difference between the effective beacon rates in the neighborhood of v ($\sigma(v)$ in Equation 5.2), i.e, maximizing the balance in the use of the channel. In this case, we have decided to use the relative standard deviation to evaluate $\sigma(v)$.

$$MAX\ \eta(v) = MAX\ \frac{\left(\sum_{j}^{j \in NN(v)} br_j\right) + br_v}{MaxQ} \qquad \eta(v) \in [0, +\infty) \tag{5.1}$$

$$MIN\ \sigma(v) = MIN \frac{\left(\sum_{j}^{j \in NN(v)} (br_j - \bar{br_v})^2\right) + (br_v - \bar{br_v})^2}{|NN(v)|} \qquad \sigma(v) \in [0, 1] \tag{5.2}$$

$\bar{br_v}$ represents the average beacon data rates in the neighborhood of v (see Equation 5.3).

$$\bar{br_v} = \frac{\left(\sum_{j}^{j \in NN(v)} br_j\right) + br_v}{|NN(v)| + 1} \tag{5.3}$$

Furthermore, the selected beacon rates computed by the algorithms br_v should not generate network congestion. Therefore, computed solutions should not exceed the effective capacity of the channel (ω) even $\eta(v)$ may return larger values, which means that the computed solutions are subjected to the restriction presented in Equation 5.4.

$$\eta(v) \leq \omega \tag{5.4}$$

The congestion control algorithms guarantee that the VANET nodes generate at least $br^{MIN} \in BR$, which is the minimum beacon rate ($br^{MIN} < br^i$, $br^{MIN}\ \forall br^i \in BR$, $br^i \neq br^{MIN}$), and never more than $br^{MAX} \in BR$, which is the maximum beacon rate ($br^{MAX} > br^i\ \forall br^i \in BR$, $br^i \neq br^{MAX}$).

Therefore, the fair beacon rate optimization problem consists in finding for each VANET node v its current br_v that optimizes both objectives with the restriction described in Equation 5.4. The computations are carried out by each node itself (there is not any central management entity) taking into account $MaxQ$, α, and the beacon rates of all its neighbors ($NN(v)$).

5.3 Greedy Dynamic Broadcasting

In this thesis, we define a set of different methods to dynamically compute optimized beacon rates to address the FBR optimization problem. These methods compose the ***FREEDY (Fair beacon Rate grEEDY)*** family of algorithms. The proposed FREEDY algorithms are fully distributed (executed periodically by each node of the VANET after a given window time), thus, no central manager entity is used.

As most of the proposed congestion control methods presented in the literature, the FREEDY's perform two main operations: **network monitoring** and **network components re-configuration** (Lochert et al., 2007). In this case, the network monitoring is performed by analyzing the actual channel occupancy (load) and the number of neighbor nodes. This information can be obtained by evaluating the queues in a given window time to measure the channel occupancy and the neighborhood.

FREEDY is composed by four different algorithms: two *Self FREEDY*, in which the nodes compute FBR using isolated information from their own network monitoring, and two *Swarm FREEDY*, which combine isolated information with information received from the neighbor nodes. FREEDY algorithms follow a cross-layer design (see Figure 5.5). The two main operations defined in FREEDY are included in the WAVE Management Entity (WME) (Campolo et al., 2015).

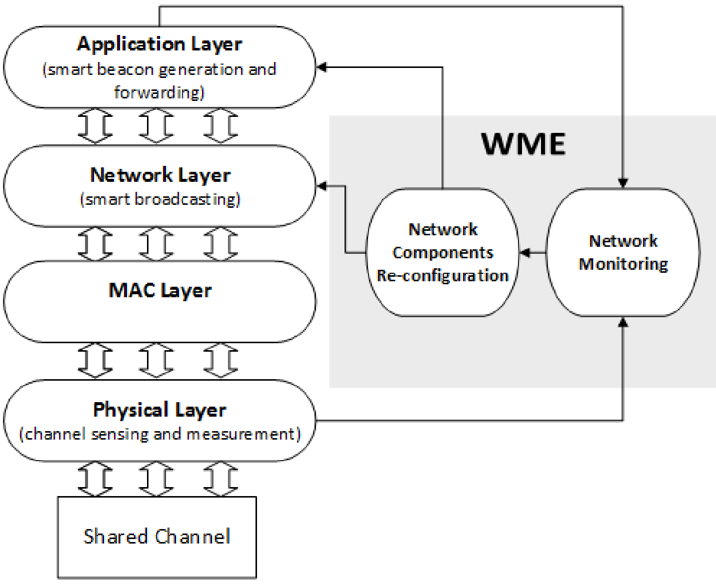


FIGURE 5.5: Global cross-layer architecture of FREEDY family.

5.4 Self FREEDY Methods

Two Self FREEDY algorithms are defined according to the information used to monitor the network and to compute the new beacon rates: ***Self o-FREEDY***, that evaluates the channel occupancy, and ***Self n-FREEDY***, that utilizes the size of the neighborhood of the node (1-hop distance nodes). Both methods present the same two main software components in their architecture (see Figure 5.6):

- *Self Queue Monitoring Component (SQMC)*: It evaluates the filling level of the queues or the number of neighbor nodes. If it is necessary, the SQMC invokes the Beacon Rate Adaptation Component.

- *Beacon Rate Adaptation Component (BRAC)*: It analyzes the information received from the other component to take the decision about which will be the new beacon rate better suited to the current network status.

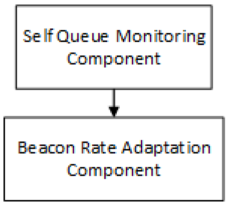


FIGURE 5.6: Main components of Self FREEDY.

5.4.1 Self *o*-FREEDY

Self *o*-FREEDY computes FBR by adapting the beacon rate to the current channel occupancy by applying a reduction to the beacon rate if the occupancy starts to be close to α. Figure 5.7 summarizes the main steps of this algorithm. Self *o*-FREEDY monitors the queue, when it detects the channel occupancy CO exceeds the channel occupancy threshold *CO_Threshold*, them *Self o-FREEDY BRAC* is invoked in order to apply the reduction function $red : [CO\_Threshold, 1] \rightarrow BR$ to compute br_v.

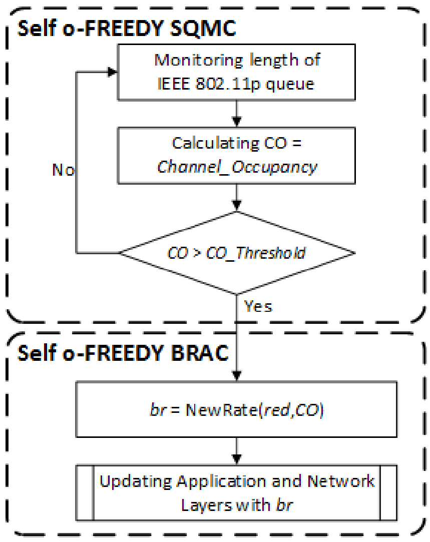


FIGURE 5.7: Complete flowchart of the Self *o*-FREEDY algorithm.

In this analysis, the use case evaluated to test beaconing adaptation is based on the real requirements of most CVS applications in the literature (Campolo et al., 2015), in which $BR \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $\alpha = 0.8$. In addition, after an initial experimentation, the $CO\_Threshold$ is set to 0.6 and *logarithmic decay* is the applied red function (see Equation 5.5).

$$br_v = \left\lceil \frac{MAX}{1 + A \times e^{B \times CO^m}} \right\rceil \tag{5.5}$$

5.4.2 Self *n*-FREEDY

The main idea of the Self *n*-FREEDY consists in dividing the maximum effective capacity of the channel into all the nodes in the neighborhood (including the own node). The algorithm starts by analyzing the received beacons in the queues to compute $|NN(v)|$, i.e., the number of neighbor nodes (see Figure 5.8). Then, the *tentative beacon rate* (tbr_v) is computed dividing the effective capacity of the channel (ω) by the number of the neighbor nodes plus one (see Equation 5.6). If tbr_v is higher than br^{MAX} (10 Hz in our studies), then br_v is equal to br^{MAX}. Otherwise, br_v is equal to tbr_v (see Equation 5.7).

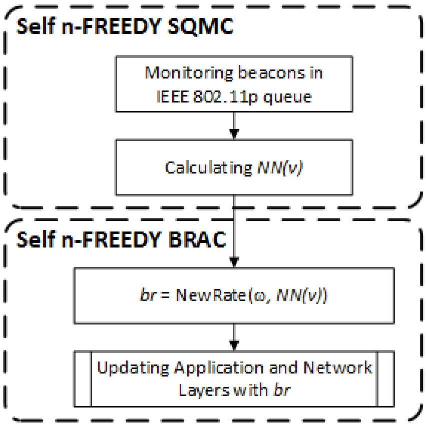


FIGURE 5.8: Complete flowchart of Self n-FREEDY algorithm.

$$tbr_v = \left\lfloor \frac{\omega}{|NN(v)| + 1} \right\rfloor \quad (5.6)$$

$$br_v = \begin{cases} tbr_v & \text{if } tbr_v \leq br^{MAX} \\ br^{MAX} & \text{if } tbr_v > br^{MAX} \end{cases} \quad (5.7)$$

5.5 Swarm FREEDY Methods

Swarm FREEDY algorithms combine self measured (monitored) information with shared congestion control data from the neighbor nodes. This control information is encoded as an integer value in the beacons to be broadcasted. This information is stored in a temporal buffer (*BRBuffer*) in order to utilize it in the near future beacon rate computations.

We have designed two different Swarm FREEDY algorithms according to the information utilized to compute the new beacon rates: *Swarm o-FREEDY*, that evaluates the channel occupancy, and *Swarm n-FREEDY*, that utilizes the size of the neighborhood.

The Swarm FREEDY algorithms have three different software components (see Figure 5.9):

- *Self Queue Monitoring Component (SQMC)*: It evaluates the filling level of the queues or the number of neighbor nodes in order to update BRBuffer.
- *Swarm Information Exchange Component (SIEC)*: It decodes information encoded in the received beacons and updates BRBuffer.
- *Beacon Rate Adaptation Component (BRAC)*: It analyzes the information stored in BRBuffer to compute the new beacon rate.

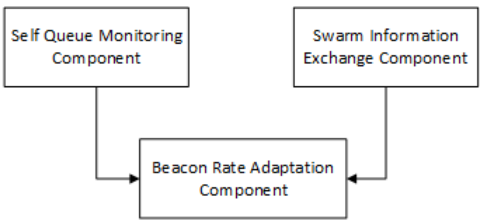


FIGURE 5.9: Main components of FREED algorithms.

In short, the SQMC and SIEC components operate in parallel with the aim of updating the information stored in the BRBuffer, and the BRAC component updates the beacon frequency br according to the stored information in the buffer. The main difference between the two defined swarm methods, Swarm o-FREEDY and Swarm n-FREEDY, is the type of information stored in the buffer and how it is used.

5.5.1 Swarm *o*-FREEDY

The main idea applied in Swarm *o*-FREEDY is: if a node detects that the CO is higher than the effective capacity of the channel (ω) in a given percentage, then all the nodes of the neighborhood have to reduce their beacon rates in the same percentage.

In this method, the BRBuffer is a vector of natural values with ten components, [x_1 x_2 ... x_{10}]. Each component x_i stores the number of petitions to modify its current beacon rate (br^t) according to Equation 5.8. Therefore, i is the multiplicative factor to be applied to compute the new beacon rates. These values are stored in the BRBuffer during a given window of time.

$$br^{t+1} = br^t \times (i \times 0.1) \tag{5.8}$$

For example, if BRBuffer=[0 0 10 25 0 0 0 0 0 1], then it means that the node has received 36 beacons that included the following broadcasting protocol information: 10 petitions to use a new beacon rate which is the 30% ($x_3 = 10$) of the current one, 25 requests to use a new beacon rate that is the 40% ($x_4 = 25$) of the current one, and one petition to not apply any modification ($x_{10} = 1$) on the current beacon rate ($br^{t+1} = 100\%\ br^t$).

Figure 5.10 summarizes the Swarm *o*-FREEDY complete procedure. Notice that SQMC and SIEC are executed in parallel and BRAC is run once just after a given *timeout* is reached.

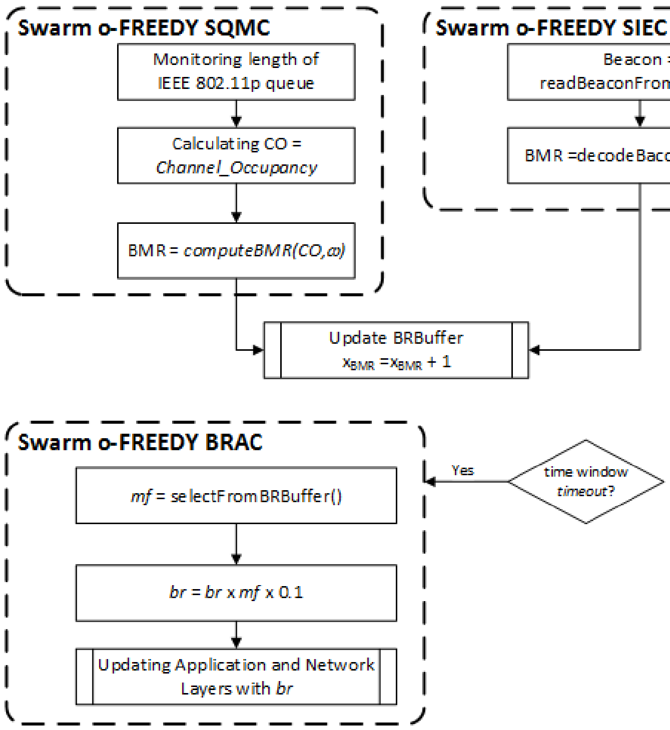


FIGURE 5.10: Complete flowchart of the Swarm *o*-FREEDY algorithm.

In the Swarm *o*-FREEDY, the SQMC component monitors the IEEE 802.11p queue length to obtain the actual CO. According to CO, it includes in the beacons to be broadcasted the *beacon modification request* ($BMR \in [1, 10]$) which is computed by following Equation 5.9. At the same time, the protocol modifies its own BRBuffer by adding an *own* petition according to the BMR computations, the BMR-*th* component of the vector is incremented ($x_{BMR} = x_{BMR} + 1$).

$$BMR = \begin{cases} 10 & CO \leq \omega \\ 10 - \left\lceil \left(\frac{CO - \omega}{\omega} \right) \times 10 \right\rceil & \omega < CO < 2 \times \omega \\ 1 & CO \geq 2 \times \omega \end{cases} \tag{5.9}$$

The SIEC decodes the beacons received in order to extract the BMR requested by the neighbors and to update the BRBuffer. The buffer is modified by increasing in each component of the vector the request received, i.e., $x_{BMR} = x_{BMR} + 1$.

The BRAC is invoked at the end of a given window time in order to compute the new beacon rate br^{t+1} according to the information stored in the BRBuffer. It applies a multiplicative factor mf to the current beacon rate according to the information stored in BRBuffer ($br^{t+1} = br^{t} \times mf \times 0.1$). There are two different variants of Swarm *o*-FREEDY depending the what metric is evaluated to select the mf to use: a) the **Swarm *o*-FREEDY-med**, which selects the median value between the two most requested values of the BRBuffer, and b) the **Swarm *o*-FREEDY-mod**, which chooses the most requested one (the mode).

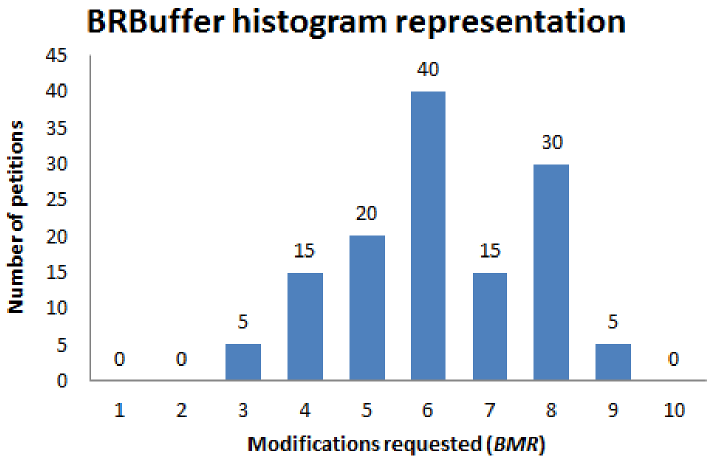


FIGURE 5.11: Histogram that represents a given BRBuffer which stores the beacon rate modification requests received.

Figure 5.11 shows a BRBuffer that stores the following information [0 0 5 15 20 40 15 30 5 0]. According to these values, Swarm *o*-FREEDY-med computes mf=7 (median value between 6 and 8) and Swarm *o*-FREEDY-mod mf=6 (the most frequent BMR).

5.5.2 Swarm *n*-FREEDY

The Swarm *n*-FREEDY is motivated by the same idea as Self *n*-FREEDY: the channel should be proportionally divided according to the neighborhood size. However, in this case, in addition to the information about the own neighbor nodes, this new algorithm utilizes the information about the number of neighbor nodes of its neighbors to increase the accuracy.

In Swarm *n*-FREEDY, the BRBuffer of each node is also a vector of natural values x_i, where each one stores the number of petitions received by the node to change its beacon rate to i beacons per second. For example, if BRBuffer=[0 0 10 25 0 0 0 0 0 1], then it means that the node has received 10 requests to change the beacon rate to 3 Hz ($x_3 = 10$), 25 to change to 4 Hz ($x_4 = 25$), and 1 to change to 10 Hz ($x_10 = 1$).

Figure 5.12 summarizes the Swarm *n*-FREEDY complete procedure. Notice that SQMC and SIEC are executed in parallel and BRAC is run once just after a given *timeout* is reached.

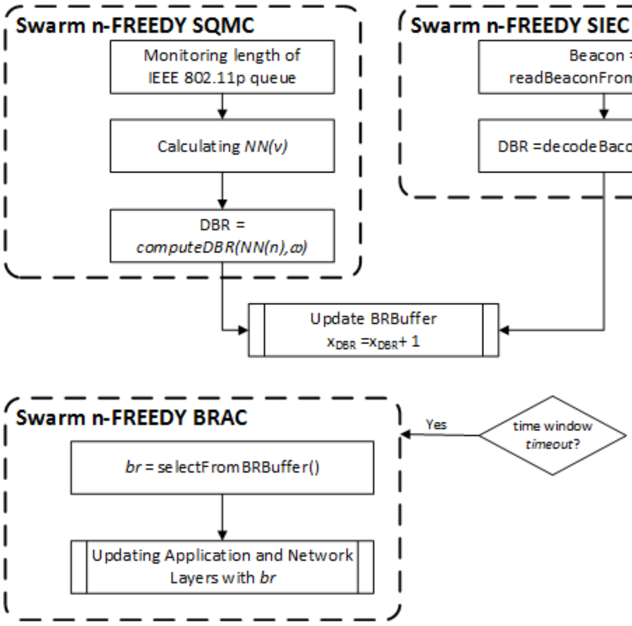


FIGURE 5.12: Complete flowchart of the Swarm n-FREEDY algorithm.

The Swarm n-FREEDY SQMC component behaves similarly to Self n-FREEDY. The difference is that the swarm method do not update the current beacon rate, in contrast, it increases the BRBuffer component of the *desirable beacon rate* (DBR) that is computed as br_v in Equation 5.7, i.e., $DRB = br_v$. Thus, this component updates the BRBuffer by increasing the DBR-th component ($x_{DBR} = x_{DBR} + 1$) and includes the DRB value in the new beacons in order to inform the neighbor nodes.

In this method, the SIEC decodes the beacons received to extract the DBR requested by the neighbors to update the BRBuffer. The buffer is modified by increasing in each component the request received, i.e., $x_{DBR} = x_{DBR} + 1$.

In addition, after a given timeout timer (window time), BRAC is executed to compute the new beaconing frequency br^{t+1} according to the DBRs stored in BRBuffer. There are two different variants of Swarm n-FREEDY depending the what metric is evaluated to select the br^{t+1} to use: **a)** the **Swarm n-FREEDY-med**, which chooses the median value between the two most requested values of the BRBuffer as new beacon rate, and **b)** the **Swarm n-FREEDY-mod**, which selects the most requested one (the mode). Following the example in Figure 5.11 (BRBuffer=[0 0 5 15 20 40 15 30 5 0]), in Swarm n-FREEDY-med it holds that br^{t+1}=7 and Swarm n-FREEDY-mod it holds that br^{t+1}=6.

5.6 Experimental Results

The experiments are carried out by using MATLAB. Each analyzed congestion control method is simulated 100 times over the same highway scenarios. We now define the highway VANET scenarios used in the experiments, we set the parameters of the Self o-FREEDY algorithm, and finally, we discuss the numerical results of evaluating the FREEDY algorithms.

5.6.1 Highway VANET Scenarios

The on-line fair beacon rate optimization is studied in a highway road of two kilometers long and six lanes (three lanes in each direction). Nine highway VANET scenarios are defined over this road by changing the mobility models in order to test the performance of the congestion control algorithms proposed here in different vehicular network status.

These scenarios are grouped in five **homogeneous** and four **heterogeneous** road traffic scenarios. In the first ones, the vehicles are assigned to a given lane randomly in which all the lanes have the same probability, and therefore, all the lanes have a very close number of vehicles. In the heterogeneous highway scenarios the probabilities are not the same. The vehicles have higher probability to be assigned to the external lanes than to the internal ones. The speed per lane increases from the external to the internal lanes, as in real highways. The distances between vehicles and the speeds are computed according to **the square law** for dry roads defined by Spanish authorities (DGT, 2015), i.e., $speed \times speed \simeq \frac{distance}{100}$. Tables 5.1 and 5.2 summarize the main characteristics of the used scenarios. The average values of the number of vehicles and the speed per lane are not shown in heterogeneous scenarios because these values are lane dependent (see Table 5.2) The simulation time for all scenarios is 100 seconds.

TABLE 5.1: Main characteristics of homogeneous highway scenarios for congestion control tests.

Road density and scenario name	Total number of vehicles	Average vehicles per lane	Average distances between nodes	Average speed
Very low density *(HM-Ld1)*	120	20	100 m	100.00 km/h
Low density *(HM-Ld2)*	160	26	75 m	86.60 km/h
Middle density *(HM-Ld3)*	240	40	50 m	70.71 km/h
High density *(HM-Hd1)*	480	80	25 m	50,00 km/h
Very high density *(HM-Hd2)*	800	100	<25 m	38.73 km/h

TABLE 5.2: Main characteristics of heterogeneous highway scenarios for congestion control tests.

Road density and scenario name	Total number of vehicles	Range of speeds
Very low density *(HT-Ld1)*	200	[40 km/h, 130 km/h]
Low density *(HT-Ld2)*	240	[40 km/h, 130 km/h]
High density *(HT-Hd1)*	320	[40 km/h, 130 km/h]
Very high density *(HT-Hd2)*	380	[40 km/h, 130 km/h]

In terms of communications, the vehicles are running CVS applications that require exchanging beacons with a frequency that ranges from 1 Hz to 10 Hz. The wireless devices utilized are configured with IEEE 802.11p with a communication range of 250 meters. In addition, following the MATLAB simulation presented in Mir et al. (2015), it is considered that the maximum size of the queues of IEEE 802.11p is 400 and α is 0.8.

5.6.2 Self *o*-FREEDY Parameterization

As defined in Section 5.4.1, the Self *o*-FREEDY algorithm utilizes the *logarithmic decay* as reduction function (red) to compute the beacon rate according to the current channel occupancy (see Equation 5.5). This function depends on two parameters that have to be set A and the logarithmic growth parameter B. In our case, we have selected a set of tentative values for both parameters $A \in \{1, 3, 5, 7, 9\}$ and $B \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. These set of values generate different logarithmic decay functions that are plotted in Figure 5.13.

Each possible configuration of the logarithmic decay function is evaluated by simulating 100 times the *middle density homogeneous* highway VANET scenario. The configuration that presents the best trade-off between the *occupancy* and the *balance* is A=3 and B=0.10.

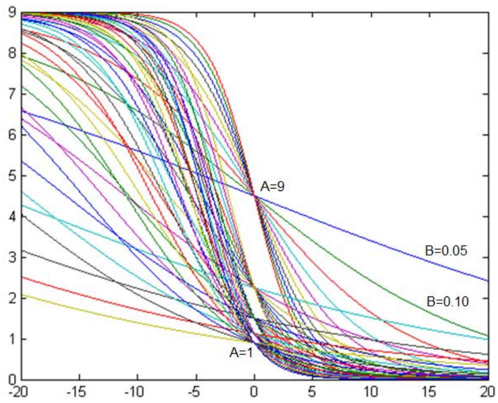


FIGURE 5.13: Different curves of the generated logarithmic decay functions.

5.6.3 Numerical Results and Discussion

This section discusses the results obtained by all the proposed FREEDY congestion control methods in the defined scenarios. In addition, other two additional methods are included in the experiments as a baseline for the comparisons:

- *Aloha based* method (Tanenbaum, 2002), the nodes broadcast beacons at the beginning of a given slot time slot without analyzing the medium, regardless whether there are another nodes using the medium or not. This is a very simple method, but the probability of having packet collisions critically grows with the size of the network.
- *CSMA* based method: When VANET nodes have to transmit a given beacon they analyze the medium. If no other node is using it then they broadcast the beacons, otherwise they drop the given beacon since, after a short while, new beacons will be generated and the information in the current beacon will be obsolete. This method reduces the likelihood of collisions.

These last two broadcasting methods utilize a fix beacon frequency. In order to analyze them, they are configured with three different fix beacon frequencies: 1 Hz, 5 Hz, and 10 Hz.

Tables 5.3 and 5.4 summarize the experimental results by showing the average and the normalized standard deviation values of the two optimized metrics, *channel occupancy* (see Equation 5.1) and *network balance* (see Equation 5.2), respectively. These results are organized in four groups: homogeneous *low density*, which includes HM-Ld1, HM-Ld2, and HM-Ld3 scenarios; homogeneous *high density*, which comprises HM-Hd1 and HM-Hd2 ones; heterogeneous *low density*, which consists of HT-Ld1 and HT-Ld2 ones; and heterogeneous *high density*, which covers HT-Hd1 and HT-Hd2 ones.

Analyzing the occupancy, there are three type of results: **a)** the ones that guarantee the proper operation of the network ($CO \leq \alpha$), **b)** the ones that may incur in a critical drop of QoS ($\alpha < CO \leq 1.0$), and **c)** the ones that exceed the channel capacity ($CO > 1.0$). In table 5.3 and 5.4 the second and third group are shaded with light and dark gray, respectively.

As expected, the **aloha based methods** allow the nodes to communicate with very low beacon rates (1 Hz) with a very poor performance. If the rate increases to 5 Hz it just is able to operate property in low density scenarios. In the other cases the channel occupancy reported value is higher than the channel capacity, which in practical means that the network is completely congested. There are no measure in terms of balance because they always broadcast packets and there are not difference between the beacon rates in the neighborhood, i.e., always $\sigma(v) = 0$ (see Equation 5.2).

TABLE 5.3: Results in terms of occupancy for each congestion control method (Equation 5.1).

Algorithms	Homogeneous scenarios				Heterogeneous scenarios			
	low density		high density		low density		high density	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Self *o*-FREEDY	0.614	3.53%	0.872	8.46%	0.642	5.85%	0.714	1.78%
Self *n*-FREEDY	0.801	5.26%	0.908	2.56%	0.836	0.71%	0.845	0.31%
Swarm *o*-FREEDY-med	0.741	14.25%	**0.721**	13.06%	0.766	17.25%	0.515	4.67%
Swarm *o*-FREEDY-mod	0.753	2.17%	0.398	25.27%	0.727	32.42%	0.263	17.50%
Swarm *n*-FREEDY-med	0.610	23.57%	0.638	5.53%	0.730	14.02%	0.691	16.48%
Swarm *n*-FREEDY-mod	0.759	2.75%	0.548	9.15%	0.736	2.78%	**0.732**	3.70%
Aloha-1Hz	0.107	29.06%	0.398	25.28%	0.142	29.29%	0.217	8.63%
Aloha-5Hz	0.537	29.02%	1.988	25.26%	0.712	29.30%	1.083	8.64%
Aloha-10Hz	1.075	29.02%	3.976	25.26%	1.424	29.25%	2.165	8.65%
CSMA-1Hz	0.108	29.02%	0.398	25.27%	0.142	29.24%	0.217	8.64%
CSMA-5Hz	0.537	28.98%	0.846	7.55%	0.694	8.88%	0.756	0.21%
CSMA-10Hz	**0.761**	2.49%	0.988	6.41%	**0.781**	5.87%	0.864	2.17%

TABLE 5.4: Results in terms of balance for each congestion control method (Equation 5.2).

Algorithms	Homogeneous scenarios				Heterogeneous scenarios			
	low density		high density		low density		high density	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Self *o*-FREEDY	0.343	28.28%	0.196	6.08%	0.278	47.14%	0.165	12.96%
Self *n*-FREEDY	0.137	83.18%	1.108	24.21%	0.285	37.20%	0.453	13.81%
Swarm *o*-FREEDY-med	0.202	47.37%	0.293	33.33%	0.249	31.93%	0.349	13.18%
Swarm *o*-FREEDY-mod	0.247	102.00%	**0.000**	0.00%	0.500	59.91%	0.268	108.63%
Swarm *n*-FREEDY-med	0.070	99.40%	0.128	23.28%	0.082	74.32%	0.105	55.17%
Swarm *n*-FREEDY-mod	**0.004**	153.26%	0.007	7.14%	**0.027**	87.12%	**0.020**	100.93%
Aloha-1Hz	-	-	-	-	-	-	-	-
Aloha-5Hz	-	-	-	-	-	-	-	-
Aloha-10Hz	-	-	-	-	-	-	-	-
CSMA-1Hz	0.000	0.00%	0.000	0.00%	0.000	0.00%	0.000	0.00%
CSMA-5Hz	0.001	667.57%	1.162	17.22%	0.102	289.00%	0.656	15.02%
CSMA-10Hz	0.520	74.90%	1.765	13.02%	0.918	21.23%	1.260	5.54%

The **CSMA based methods** improve the performance of the aforementioned non-adaptive method. Taking into account just the channel occupancy metric, they offer the best values in low density highway scenarios when CSMA-10Hz is used. However, it incurs in a very high cost in terms of balance, and therefore, it suffers from the largest differences between the amount of data that each node in the neighborhood broadcasts. This means that although the channel is efficiently used, when managed by CSMA-10 Hz the VANET applications cannot property operate because many nodes do not broadcast their beacons.

Regarding to the **Self FREEDY** methods, the channel occupancy of Self *n*-FREEDY (that divides the effective channel capacity among the neighbors) has exceeded the α value defined in our experiments. Therefore, it provokes a drop in the QoS of the network. This is because the computations take into account just the information of their neighborhood (1-hop nodes), and ignore the rest of the nodes in the same cluster. In addition, the fairness between the nodes (balance) is the least competitive of all FREEDY methods. In contrast, the Self *o*-FREEDY provides more conservative beacon rates (lower channel occupancy), although the channel occupancy surpass the α value in high density scenarios. Besides the balance provided is very poor in comparison with the Swarm FREEDY methods.

Analyzing the **Swarm FREEDY** algorithms, they provided the best trade-off results in terms of occupancy and fairness. Specifically, if we compare Swarm n-FREEDY and Swarm o-FREEDY methods, the first ones outperform the second ones in terms of occupancy in the low density scenarios, but the opposite occurs in high density scenarios. If the analysis is in terms of balance, in general, the Swarm n-FREEDY algorithms generate a more balanced network. Although, in the homogeneous high density scenarios, Swarm o-FREEDY-mod presents $\sigma(v) = 0$, which means that all nodes in the clusters broadcast the beacons with the same rate. This is due to the reduced beacon rates generated and the very low channel occupancy (the lowest for these scenarios, 0.398). Taking into account the metric used to select the values from the BRBuffer: median or mode, we can observe an emergent behavior. The algorithms that use the mode (the most repeated value in BRBuffer) the balance between the nodes is significantly higher than when they utilize the median. This means that the vehicles broadcast beacons with similar beacon rates, which is the desirable behavior.

These results are confirmed by the statistical test applied to the simulation results (Friedman and Wilcoxon tests). Tables B.15, B.16, and B.17 in Appendix B show the statistical test results. According to these tests, the **Swarm *n*-FREEDY-mod** is significantly the most competitive method in terms of both analyzed metrics.

Finally, in the light of the results in the tables we can conclude that the Swarm FREEDY algorithms present a set of methods that provide the best trade-off occupancy channel and fairness between the nodes than the other studied methods. Specifically, **Swarm *n*-FREEDY-mod** demonstrates the best performance taking into account both analyzed metrics. In addition, the two Self FREEDY broadcasting methods improve the other two non-adaptive (Aloha and CSMA based) algorithms included in the comparison.

5.7 Conclusions

This chapter has tackled the fair beacon rate optimization problem in order to address the congestion control problem of CVS applications in VANETs. Thus, a set of greedy dynamic broadcasting methods have been proposed to be applied in vehicular communications. These methods are based either on isolated information monitored by the own node (Self FREEDY) or on combining the monitored information with the shared one received in the same beacons broadcasted by the neighbor nodes (Swarm FREEDY).

In order to evaluate these methods, they have analyzed a number of highway VANET scenarios that represent a big set of diverse vehicular scenarios. According to the experimental results, the Swarm FREEDY algorithms are the most competitive ones in terms of channel occupancy and fairness (balance). Specifically, the Swarm n-FREEDY-mod method experimented the best trade-off between these two metrics. In addition, Self FREEDY algorithms outperformed other broadcasting base methods (Aloha and CSMA based broadcasting algorithms).

The Swarm FREEDY methods have demonstrated to provide a reliable and non-complex broadcasting method. They just require to include a natural value from 1 to 10 in the beacon as control information for the broadcasting algorithm (i.e., four bits long).

The Swarm FREEDY algorithms presented in this chapter are a starting point for developing new dynamic swarm broadcasting methods that utilize modern NC (e.g., Ant Colony Optimization or Artificial Bee Colony based algorithms) in order to compute more reliable beacons frequencies for VANETs.

Chapter 6

Natural Computing for Smart Roadside Unit Placement

THE deployment of the RSU infrastructure along the roads is one of the main issues that VANET designers should address. They have to decide the type and the location of the RSUs in order to maximize the service offered and minimize the deployment costs. This is even harder when considering city-scaled areas to deploy the RSU infrastructure. The current literature shows that the hard-to-solve optimization problem associated cannot be efficiently solved using classic exact methods for such instances. In this chapter, we define a novel explicit multi-objective problem which is addressed with a parallel EA. Besides this, we have defined two different greedy constructive heuristics based on the state-of-the-art ones. The chapter begins introducing the problem, after that it is mathematically formulated. Then an experimental analysis is performed taking into account real data traffic from the city of Málaga (Spain) an real wireless devices.

6.1 Introduction

This chapter focus on a specific type of agents in the VANET architecture, the RSUs. As introduced in Section 2.2.1, RSUs are devices usually installed along the roads (either using specific VANET elements or equipping already established infrastructure elements). They perform three main important functions in the VANET via V2I (see Figure 6.1): **i)** they act as information source or receiver, **ii)** they extend the effective communication range of the other nodes of the VANET, and **iii)** they provide nodes without V2B interfaces with Internet connectivity. Therefore, the deployment of a fixed infrastructure of RSUs has been treated in the domain of vehicular networks.

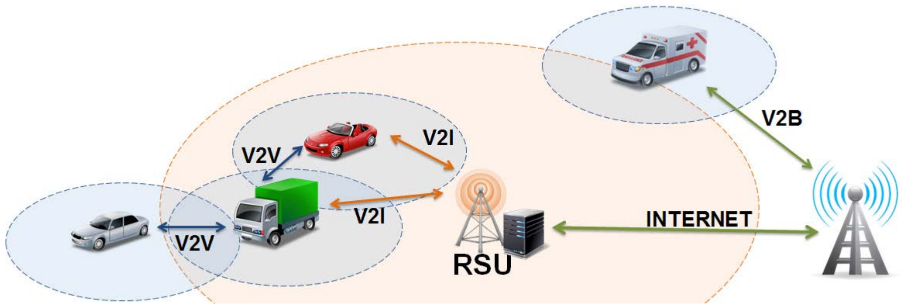


FIGURE 6.1: Global VANET architecture using a RSU.

Deploying such an infrastructure is a challenge in modern cities, because designers must decide about the number, the type, and the location of the RSUs in order to maximize QoS of the vehicular network, while satisfying and/or minimizing the deployment cost constraints. This is known as the *RSU deployment problem* (RSU-DP), which consists in placing a set of RSU terminals (antennas) along the roads of a given area, maximizing the network capabilities and minimizing the deployment costs.

The RSU-DP is a hard-to-solve optimization problem when dealing with (large) scenarios on city-scaled areas, since the number of possible solutions (i.e., sets of RSU locations) becomes very large (virtually impractical) (Reis et al., 2014). Therefore, as it happens with the previous optimization problems, traditional exact methods are not able to find accurate solutions in reasonable computation times. As presented in Section 3.3.3 some authors have proposed specific heuristics to deal with RSU-DP for limited sized instances (Ben Brahim et al., 2014; Wang et al., 2014; Patil and Gokhale, 2013; Trullols et al., 2010). In this case again, NC raises as promising strategy to address the RSU-DP, because this type of methods allows computing *good* infrastructure designs (with satisfactory QoS and reduced cost) in limited execution times (Cavalcante et al., 2012; Cheng et al., 2013).

In this chapter, we propose a novel explicit multi-objective formulation of the problem in order to overcome the drawbacks of the previously applied NC that utilize a mono-objective formulation. The multi-objective approach analyzed here is tackled by applying NSGA-II (Deb et al., 2002) in order to optimally design the RSU infrastructure within a city-scaled road network in Málaga (Spain). Aiming at obtaining realistic results, the studied scenarios consider real information about road traffic (traffic flows and road map) and hardware (network capabilities and costs). Besides this, a specific QoS model is proposed in our study: considering the number of vehicles, speed, and the coverage of street segments in the city; and a Monte-Carlo simulation approach is used to compute the corresponding QoS metric. The research work carried out here is an extension of the analysis presented in Massobrio et al. (2015a).

6.2 The RSU Deployment Problem

One of the main contributions of this work is the new multi-objective formulation of RSU-DP. The mathematical definition of this problem considers the following elements:

- A set of *road segments* $S = \{s_1, s_2, \ldots, s_n\}$, which are potential locations for placing a set of RSUs $R = \{R_1, R_2, \ldots, R_q\}$ along the city streets. Each segment s_i is defined by a pair of points (p_j, p_k), with $p_j, p_k \in P = \{p_1, p_2, \ldots, p_m\}$. Each point p_j is identified by its geographical coordinates (latitude, longitude), that coincide with corners of the streets or point-of-interest for the neighborhood. The length of a given segment s_i is given by the function *len*: $S \to \mathbb{R}^+$. RSUs can be placed at any location within each segment $s_i \in S$.
- An estimation of the number of vehicles per time period across each segment s_i, given by function NV: $S \to \mathbb{N}^+$, and the average vehicle speed for each segment, given by function *sp*: $S \to \mathbb{R}^+$.
- A set of RSU types $T = \{t_1, \ldots, t_k\}$. Each RSU type is characterized by a given deployment cost and a coverage determined by the transmission power and the antenna gain. The type of a RSU is given by the function *type*: $R \to T$.
- A cost function C: $T \to \mathbb{R}^+$, where $C(t_h)$ indicates the monetary cost of placing a RSU of type t_h in the deployed infrastructure.

Solutions of the problem are defined by a set of RSUs placed over the road segments of the city, i.e., $sol = \{R_1, R_2, \ldots, R_l\}$, where l is the number of RSUs in sol. The segments covered by a RSU are given by the function *cov*: $R \to S$, and the portion of segment s_i covered by RSU R_j is given by the function *cp*: $R \times S \to [0, 1]$.

The multi-objective version of the problem proposes to find a set of locations and the type of RSU to deploy in each location, with the goal of maximizing the *service time* given by the whole RSU infrastructure (see Equation 6.1), while simultaneously minimizing the *total cost* of deployment (see Equation 6.2). The service time is given by the number of vehicles attended by RSUs and the time they are served (considering the coverage and average speed per each road segment). Formally, the problem is defined as the simultaneous optimization of the following two objective functions:

$$\max \sum_{R_j}^{R_j \in sol} \sum_{s_i \in cov(R_j)} NV(s_i) \times \frac{cp(R_j, s_i) \times len(s_i)}{sp(s_i)} \tag{6.1}$$

$$\min \sum_{R_j}^{R_j \in sol} C(type(R_j)) \tag{6.2}$$

6.3 Implementation Details

The RSU-DP problem is treated by using NSGA-II, because as we indicated earlier in this thesis it is a state-of-the-art MOEA that has been successfully applied in many areas. Therefore, we have selected it as a baseline for the research work performed in this chapter. The NSGA-II algorithm proposed has been tailored to compute accurate solutions for the RSU-DP.

As it has been carried in our last studies in off-line optimization problems (see sections 4.4 and 4.5), a master-slave parallel model is applied to reduce the execution time demanded by the NSGA-II to evaluate the objective functions for each individual in the population. The main implementation details are presented next.

6.3.1 Problem Encoding

Solutions are represented as vectors of real numbers, having $n = \#S$ elements (the number of road segments in S). Each position on the vector defines the RSU information for the corresponding segment. The RSU type is given by the integer part of the real number (0 stands for no RSU placed on the segment, and integers $1 \ldots k$ represent RSUs of types $t_1 \ldots t_k$, respectively). The position within the segment is given by the fractional part of the real number, mapping the interval $[0, 1)$ to points in the segment $[p_j, p_i)$.

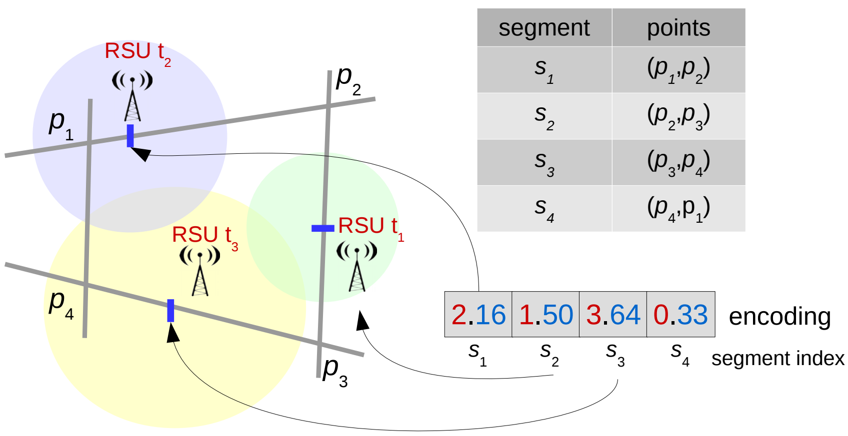


FIGURE 6.2: Solution encoding for RSU-DP.

Figure 6.2 presents an example of solution encoding for a scenario with four segments ($n = 4$) and three RSUs. The value 1.50 in position 2 of the solution vector ($s_2 = 1.50$) means that a RSU of type t_1 (1.50) is placed at the middle ($1.50 \times len(s_2)$) of the segment $s_2 = (p_2, p_3)$.

6.3.2 Objective Functions Computation

RSU-DP optimization problem has two objectives maximizing the RSU service time (as QoS indicator) defined in Equation 6.1 and minimizing the total deployment cost (see Equation 6.2). The calculation of the total cost is straightforward, by adding the cost (according to the corresponding type) of each RSU placed in the scenario.

For computing the QoS metric, we consider the distances and values depicted in the example illustrated in Figure 6.3: the RSU placed in the point "×" in segment $s_1 = (p_1, p_2)$ covers the subsegments c_1 (in s_1), c_2 (in s_2), both in street A, and c_3 (in s_3), and c_4 (in s_4) in street B, according to the coverage defined by the RSU type. The number of effective vehicles attended in this example is given by the following equation:

$$\sum_{i=1}^{i=4} NV(s_i) \times \frac{c_i}{sp(s_i)} \tag{6.3}$$

This operation requires computing the intersections between the road segments and the circle representing the coverage of the RSU. Coverage is computed using a Monte-Carlo simulation approach: each segment is divided in 10 points and the length of the subsegment c_i is computed by simulation, considering the coverage radius of the corresponding RSU.

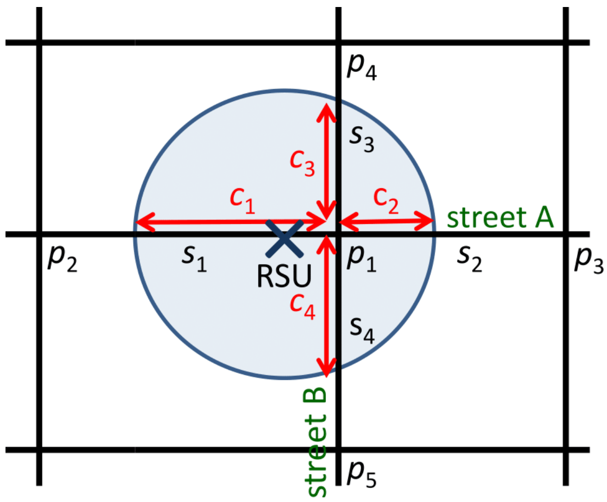


FIGURE 6.3: Calculation of the vehicles attended by a RSU.

Given that the distances involved in the problem are relatively small, we use the Euclidean distance in the latitude-longitude space as an estimation, with negligible error. This approximation makes the problem significantly faster to compute, thus improving the overall performance of the algorithm. Since the distance of a degree of longitude depends on the latitude, it is necessary to adjust for that by multiplying the longitude by the cosine of the latitude.

6.3.3 Evolutionary Operators

In this study, we have design specific evolutionary operators in order to improve the efficacy of the NC algorithm applied. These specific evolutionary search operators are described next.

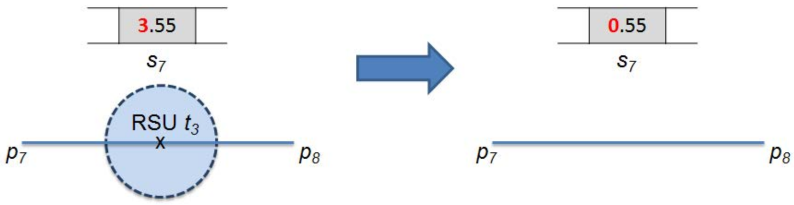
Initialization

The population is initialized by considering the solutions computed by two randomized greedy heuristics for the problem (see a description of the greedy heuristics in Section 6.4.2). A total number of 20% of the individuals in the population are seeded using the greedy solutions. Given that one of the extremes of the ideal Pareto front is known (i.e., the solution that places no RSU has cost 0), we add that solution to the initial population as well. The remaining individuals of the population are randomly initialized, using reals from the interval [0, $k + r$] being k the number of different RSU types in T, and $r \in [0, 1)$.

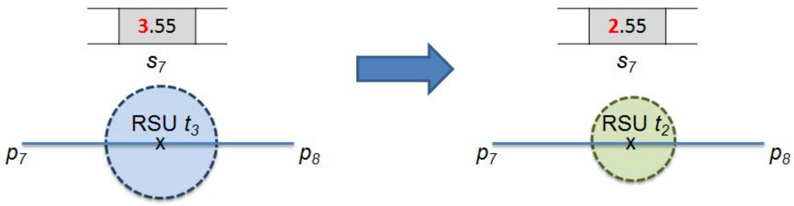
RSU-DP Mutation

An ad-hoc mutation operator is designed to provide diversity to the search, which works as follows. Mutation is applied over solutions with probability p_M. When applied, the mutation operator selects a number of segments to modify (s_i) according to a uniform probability. Then, it applies a variation over each s_i segment as follows:

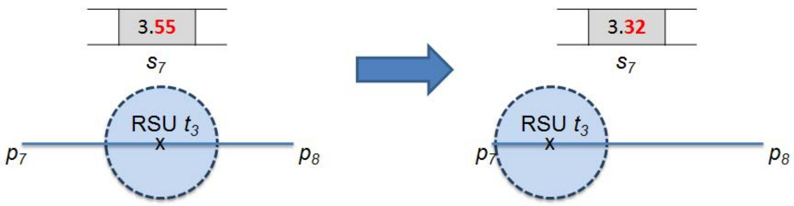
- with probability π_A, the mutation operator sets the integer part of the selected gene value to 0, thus removing the RSU (if any) from the corresponding segment (see Figure 6.4.a);
- with probability π_B, the mutation changes the type of the RSU (if any) to a random type picked uniformly from set T, thus changing the type of the RSU (or adding one if there was none) (see Figure 6.4.b);
- and with probability $1 - \pi_A - \pi_B$, a *Gaussian mutation* on the value for segment s_i is applied with a standard deviation given by parameter σ in order to change the position of the RSU within the segment (see Figure 6.4.c).



a) Removing RSU (applied with probability π_A).



b) Changing RSU type or adding RSU (applied with probability π_B).



c) Changing RSU position (applied with probability $1 - \pi_A - \pi_B$).

FIGURE 6.4: Graphical representation of RSU-DP mutation operator.

Recombination

The crossover operator is *Intermediate Recombination*; two parents $\vec{x} = \{x_i\}$ and $\vec{y} = \{y_i\}$ are combined to generate two offspring O1 and O2; they satisfy $O1_i = \alpha_i x_i + (1 - \alpha_i) y_i$ and $O2_i = \beta_i y_i + (1 - \beta_i) x_i$ with α_i, β_i randomly chosen from the interval $[-\mu, 1 + \mu]$ for a given value of parameter $\mu \in [0, 1]$. The recombination operator is applied with a probability p_C.

6.4 Experimental Results

This section presents the experiments carried out to solve RSU-DP optimization problem by using the proposed MOEA, which is implemented by using *ECJ Java-based Evolutionary Computation framework* (White, 2012). Further details about this experimental analysis can be found in Massobrio et al. (2015b).

6.4.1 Problem Instances

In order to apply our evolutionary approach, we define a real world problem instance based on a real map of Málaga, real road traffic data, and real wireless antennas to equip the RSUs.

The map covers an area of 42.557 km^2 in the city, including a number of 106 points, which define 121 segments with lengths between 55 and 1556 m (see Figure 6.5.a). All major traffic ways, including avenues and important streets in Málaga are sampled. Some important avenues with large traffic volume define multiple segments in the map (e.g., *Avenida de Andalucía*, *Avenida de Velázquez*, *Avenida de Valle Inclán* and *Paseo Marítimo Pablo Ruiz Picasso*, all of them with more than six segments defined in the map).

The traffic data was collected by the Málaga governmental institutions using a set of sensors located along the roads. These sensors returned the total number of vehicles that circulated during the last three months of 2014. Thus, this information is utilized to define the *normal* pattern for traffic (see Figure 6.5.b). In addition, two probabilistic multiplicative factors are applied over the *normal* pattern to define two other ones: *low* pattern, reducing the traffic randomly in [0%–20%] and *high* pattern, increasing the traffic randomly in [0%–20%]. These patterns represent situations with lower and higher road traffic density, respectively.

The RSUs hardware is defined by a processing unit equipped with a IEEE 802.11p network interface. Each network interface is connected to an external antenna to improve the communication range according to a given antenna gain. The gain, measured in decibels (dBi), is a measure of the power of the radio signal radiating from the antenna. Generally, the higher the gain of an antenna, the longer radio range will be obtained. The used antennas have to operate in 5.9 GHz band utilized in IEEE 802.11p standard. Our instance includes three types of RSUs that differ in the antenna gain connected. Three types of IEEE 802.11p antennas are considered, according to three commercial omni-directional antennas that can be found in *Cetacea on-line shop* (Cetacea, 2015). Table 6.1 summarizes the main features of such antennas.

In order to define the *effective radio range* (ERR) of each RSU, we evaluate via simulations the average PDR, at different distances (from 0 to 650 m) for each RSU. The experiments were performed using the ns-2 simulator (NS2, 2015) to simulate vehicular communications using IEEE 802.11p PHY/MAC standard in a urban scenario defined in a one lane road of 1 km with one RSU and 10 moving cars at 40 km/h. During the simulations, the RSU sent continuous data streams at 256 Kbps to the cars. The Probabilistic Nakagami radio propagation model (Saunders and Aragon, 1999) is used to represent channel fading characteristics of urban scenarios. The results of these experiments are summarized in Figure 6.6.

a) Segments that represent the road map.

b) Road segments in which their width represents the traffic density.

FIGURE 6.5: Road information from Málaga taken into account in RSU-DP analysis.

Finally, in order to ensure a competitive QoS, we defined the ERR of each RSU as the distance at which the average PDR is equal or higher than 66.67%. Therefore the ERR is set for each antenna as it is shown in Table 6.1.

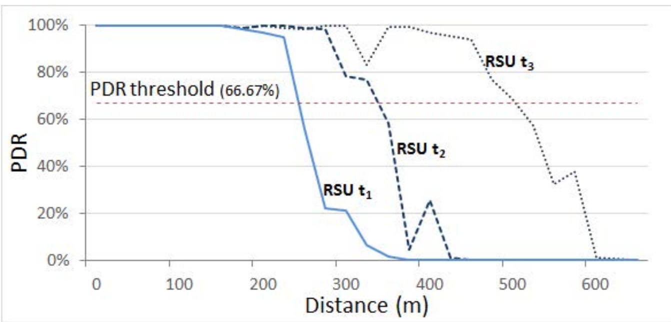


FIGURE 6.6: PDR results regarding to the three types of antennas analyzed in RSU-DP.

TABLE 6.1: General information about the used antennas to address RSU-DP.

type	*commercial model*	*gain*	*ERR*	*cost*
t_1	Echo Series Omni Site Antenna	6 dBi	243.12 m	121.70 $
t_2	Echo Series Omni Site Antenna	9 dBi	338.70 m	139.20 $
t_3	Echo Series Omni Site Antenna	12 dBi	503.93 m	227.50 $

6.4.2 Comparison Against Two Greedy Strategies

In order to compare the results achieved by the proposed MOEA, we develop two randomized greedy heuristics, focused on each one of the problem objectives. These heuristics apply intuitive ideas that simulate the behavior of human-planning strategies, and they are improved versions of the methods defined by Trullols et al. (2010) and later used in the comparative study by Cavalcante et al. (2012). The improvements in our heuristics (over the ones in Trullols et al. (2010)) include: **i**) in our methods, RSUs can be located anywhere within road segments (instead of placing RSUs only at road intersections), **ii**) we consider a variable number of RSUs (instead of using a fixed number of RSUs), and **iii**) a set of RSU types and coverages are considered (instead of a single RSU type). We compare the results achieved by the proposed MOEA against two greedy heuristics.

The two greedy heuristics are briefly described next:

1. *Greedy QoS* (***GQoS***): the set of segments P is sorted according to the QoS they provide (i.e., the ratio between number of vehicles and average speed) in case they are totally covered by a RSU. Iteratively, GQoS adds to the solution the RSU that provides the best QoS (or cheaper in case of overlapping), at a random location in the sorted segments, while computing the segments covered by the located infrastructure in each step. Segments that are already covered are not taken into account to be included in the solution.
2. *Greedy cost* (***GCost***): starting from the solution computed by GQoS, the algorithm tries to reduce the cost without significantly affecting the quality of service. Different solutions are explored, by replacing existing RSUs by cheaper ones, or deleted, and the option with the lower QoS degradation is selected. The algorithm stops when all segments are considered or when the QoS of the solution is equal to $\alpha \cdot Q$ where Q is the best QoS value achieved by the greedy algorithm for QoS and $\alpha \in [0, 1]$. For the experimental analysis, GCost was executed using $\alpha \in \{0.70, 0.75, 0.80\}$.

Parameter Settings of NSGA-II

We perform an analysis to find the best values for NSGA-II parameters. In the parameter setting experiments, the best results are obtained using the configuration: *population size* = 72, $p_C = 0.95$, $p_M = 0.01$, $\pi_A = 0.5$, $\pi_B = 0.25$. The value of μ in the *Intermediate Recombination* operator is set to 0.25. In the Gaussian mutation the value of parameter σ is 0.25.

6.4.3 Numerical results

The experimental analysis is oriented to evaluate the problem solving capabilities of NSGA-II for the RUS-DP. On the one hand, we compare NSGA-II with the greedy heuristics; on the other hand, we evaluate several standard multi-objective optimization metrics (Deb, 2001): generational distance (GD), to evaluate the solution quality; spread (I_Δ), to evaluate the distributions of solutions; and the combined metric *relative hypervolume* (R_{HV}), to evaluate both quality and dispersion. We also analyze the Pareto fronts computed by NSGA-II for each scenario in the experimental evaluation. For each problem instance, we perform 30 independent runs of the MOEA and both greedy algorithms.

In the experimental analysis the proposed MOEA shows a good solving capability. NSGA-II significantly outperforms the two greedy heuristics while computing accurate Pareto fronts. The solutions computed by the greedy heuristics tend to group in different areas of the solution space, depending on the parameters used for their execution. Therefore, the results obtained by the MOEA are compared against the average results of each group of greedy solutions.

The improvements of NSGA-II over the greedy strategies are reported in Table 6.2. The selected NC algorithm is able to improve the QoS of the greedy heuristics in up to **6.0%** while keeping the same cost, and improve up to **37.1%** the cost of the greedy heuristics while keeping the same QoS (this value represents a $5218.4 saving on a $14079.7 investment). Regarding the cost objective, NSGA-II improves over the greedy results 19.8% in average (for low traffic instance), 20.3% in average (for normal traffic instance), and 17.0% in average (for high traffic instance). Improvements on QoS are smaller but still significative: 4.2% in average (for low and normal traffic instances) and 3.5% in average (for high traffic instance).

TABLE 6.2: NSGA-II improvements over greedy heuristics in solving RSU-DP.

Instance	*Cost improvement (%)*		*QoS improvement (%)*	
	Best	*Avg.±Std.*	*Best*	*Avg.±Std.*
low	37.1	19.8±10.3	5.6	4.2±0.9
normal	36.9	20.3±10.7	6.0	4.2±1.2
high	31.0	17.0±8.3	5.5	3.5±1.4

Figures 6.7 and 6.8 illustrate two different solutions (RSU deployments) with the same QoS computed by the EA and the the heuristics, respectively. As it can be seen, the reduction of costs achieved by the NC method is principally because the RSUs are located such that they avoid unnecessary network overlapping. Thus, they can offer similar coverage by using a smaller amount of antennas.

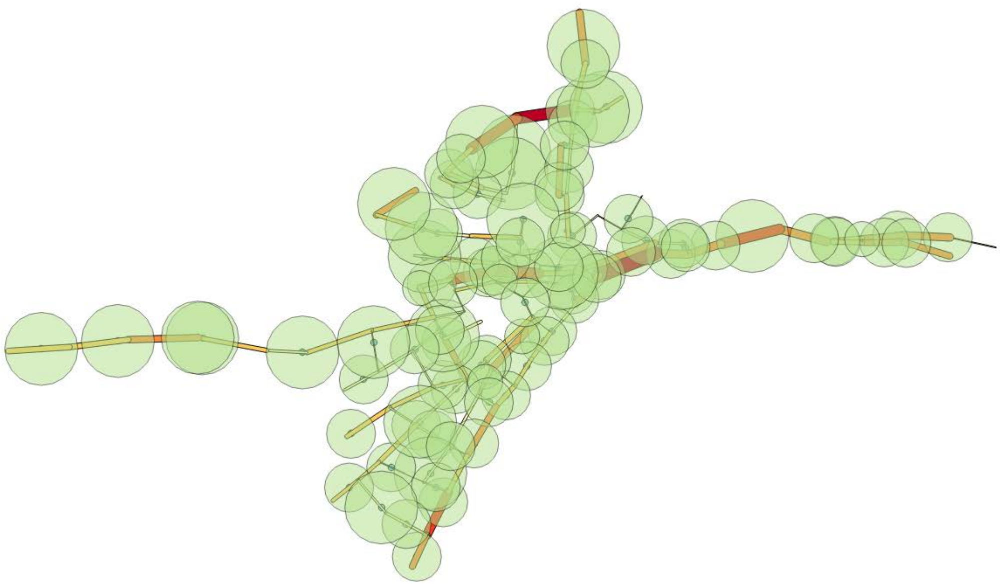


FIGURE 6.7: RSU-DP deployment computed by applying NC.

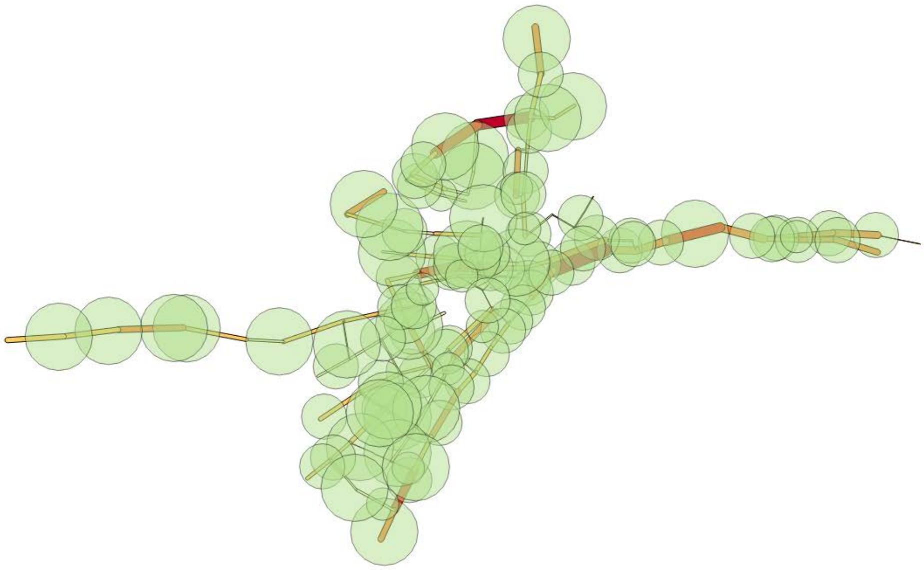


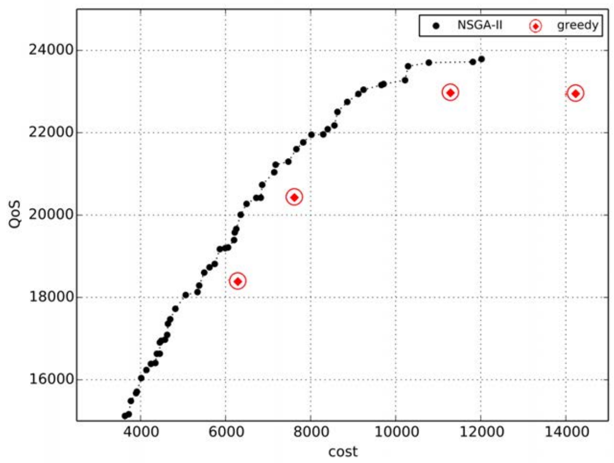
FIGURE 6.8: RSU deployment computed by using a heuristic method.

Table 6.3 shows the average, standard deviation and best results for the studied standard multi-objective optimization metrics. The ideal Pareto front (which is unknown for the problem instances studied) is approximated by gathering the non-dominated solutions obtained over all executions performed. The small generational distance values indicate a good convergence to an hypothetical ideal Pareto front, and demonstrate the robustness of the NSGA-II approach. Spread values suggest a good distribution of the non-dominated solutions, which is similar for the three analyzed instances. These results are confirmed by the unitary value of the relative hypervolume metric.

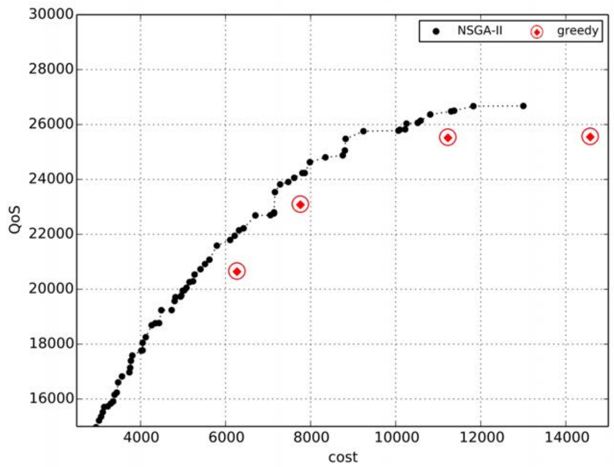
TABLE 6.3: NSGA-II results (multi-objective optimization metrics) when solving RSU-DP.

Instance	*GD*		I_Δ		R_{HV}	
	Best	*Avg.±Std.*	*Best*	*Avg.±Std.*	*Best*	*Avg.±Std.*
low	1.2	1.5±0.3	0.7	0.7±0.0	1.0	1.0±0.0
normal	1.2	1.5±0.1	0.7	0.7±0.0	1.0	1.0±0.0
high	1.0	1.6±0.2	0.7	0.7±0.0	1.0	1.0±0.0

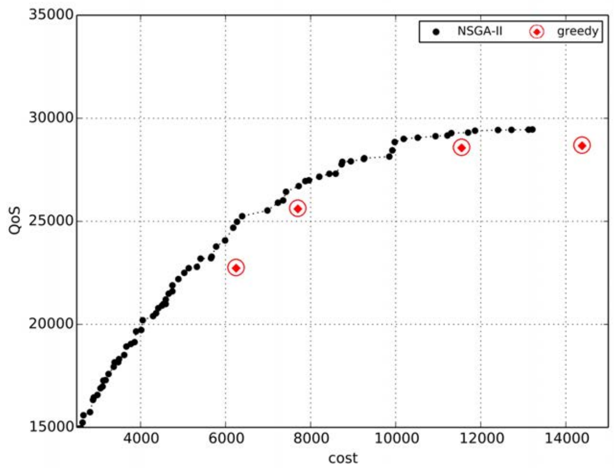
Finally, figures 6.9.a, 6.9.b, and 6.9.c illustrate the global Pareto fronts achieved by NSGA-II in the 30 executions performed, compared against the results obtained by the state-of-the-art greedy algorithms on low, normal, and high traffic density scenarios. In these figures, black dots represent the non-dominated solutions that belong to the obtained Pareto fronts and the red circled points are the solutions computed by the greedy algorithms. As it is illustrated, the solutions of the computed Pareto fronts by the parallel EA dominate the ones of the greedy algorithms. Therefore, the heuristics compute RSU designs with lower QoS and higher deployment costs than our NC method.



a) Low traffic density.



b) Normal traffic density.



c) High traffic density.

FIGURE 6.9: Global Pareto fronts computed by the NSGA-II and the solutions obtained by the greedy algorithms in solving RSU-DP.

6.5 Conclusions

An explicit multi-objective formulation is presented and a parallel MOEA is applied to solve real problem instances in city-scaled scenarios.

In the experimental analysis, the proposed MOEA shows good problem solving capabilities. NSGA-II significantly improves over two greedy heuristics for the problem (which are improved versions of methods proposed and used in the literature). The NSGA-II improvements are up to **37.1%** (19.0% in average) in the cost objective and **6.0%** (5.7% in average) in the QoS objective. Additionally, NSGA-II is able to compute accurate Pareto fronts, providing different trade-off solutions for the problem. The novel analysis presented in this chapter represents a first successful step to address RSU-DP in large-scaled cities by using NC.

Chapter 7

Real World VANET Experiments

THE evaluation of vehicular networks is overwhelmingly carried out in the present literature with simulations. The degree of realism of those is limited because their computations simplify the real world interactions too much in many cases. In this chapter, we define two different outdoor VANET testbeds to evaluate the performance of short range vehicular communications. This study is carried out to confirm the efficiency of configuring nodes with protocols improved by applying NC and to evaluate the use of lightweight personal devices in V2V communications. This chapter introduces the importance of performing outdoor experiments and details the experimental analysis results.

7.1 Introduction

The different solutions obtained in the previous chapters have been evaluated by using simulators. This analytic method is limited by the complexity and dimension of real world systems, which usually require simplifications and approximations that generally lead to differences between their results and real world behavior.

As a useful complement (or even realistic substitute) for simulations we can use experimental real world testbeds. Testbeds have important advantages with respect to the simulations because these tests are carried out in a real world environments offering close-to-real or real performance, as well as revealing behavioral issues (Pinart et al., 2008). However, there is a lack of scientific articles that use outdoor experiments in the field of vehicular networks. The main reasons for this may be the unavailability of resources (vehicles and road equipment), the difficulties in doing field studies, and the accuracy of the performance analysis.

Despite these limitations some authors have analyzed the feasibility of VANET communications by using laptops equipped with IEEE 802.11bg wireless interfaces (Gass et al., 2006; Bychkovsky et al., 2006; Lee et al., 2007). In Festag et al. (2004), the authors analyzed FleetNet's platform, which combines IEEE 802.11bg (WLAN) and GPRS (cellular) wireless interfaces. Recently, the performance of IEEE 802.11p PHY standard has been studied via V2I communications (Paier et al., 2010; Mangel et al., 2011). As the availability of IEEE 802.11p devices is very limited at present, other studies have utilized IEEE 802.11a PHY standard, which uses the band (5 GHz) closest to the one used by IEEE 802.11p (Sanchez et al., 2014).

This chapter presents two different experimental analysis of vehicular communications performed in real outdoor VANETs (real vehicles and wireless devices): **1)** we analyze the VDTP protocol (optimized versus standard) in order to validate the results of Chapter 4; and **2)** we evaluate the performance of V2V communications by using personal devices (smartphones, tablets, and laptops) and two different PHY standards (IEEE 802.11g and IEEE 802.11a), in order to study the possibility of deploy VANETs without installing specific OBUs.

7.2 Performance Analysis of Improved VDTP in Real World Tests

In the present section, we are aimed at defining a testbed in order to study the performance of the VDTP file transfer service between cars in a real urban VANET. In these outdoor experiments, the VDTP protocol has been tested following different parameter settings: the optimized VDTP configurations proposed in Chapter 4 and the standard one proposed by CARLINK experts (Luna S., 2008). Thus, the results offer the possibility to confirm the QoS improvements on a VANET's performance when optimized protocols are used, validating the previous results obtained through simulations.

7.2.1 VDTP Testbed Definition

The VANET scenario utilized in our experiments is comprises of two cars moving through the roads in an area of 1.44 km^2 from the downtown of Málaga, Spain. The roads are opened to the general traffic during a non rush hour, so the number of vehicles traveling through our scenario are not constant (see Figure 7.1). Therefore, the speed and the distance between the vehicles vary over time, just as it would be in any real city. According to the tracking information, during the experiments the average distance between the nodes is 77 meters.

FIGURE 7.1: Nodes during the real world VDTP experiments (*P=petitioner* and *O=owner*).

Regarding the communication platform, the cars are equipped with a laptop with a *PROXIM ORiNOCCO PCMCIA (IEEE 802.11bg)* Wi-Fi transceiver (Proxim, 2015) connected to a 7 dBi omnidirectional antenna located on roof top of the car (see Figure 7.2). The file transfers are performed by using the six different configurations for urban VDTP shown in Table 4.4. Five of these configurations have been obtained automatically by using NC (i.e., PSO, DE, GA, ES, and SA) and the other one was proposed by Luna S. (2008), in this study it is named *EXPERTS* configuration. The global network is configured following the VANET specifications used in the simulations presented in Chapter 4. Additionally, we included a GPS unit in each vehicle to track their movement.

In order to perform a general study, we take into account five types of data files of different sizes: 100 kBytes and 500 kBytes typical in traffic information services; and 1 MByte, 5 MBytes, and 10 MBytes that contain multimedia content. For each VDTP configuration, the vehicles exchange 15 files of each type, i.e., for each VDTP configuration there are 75 (5×15) file transfers. In turn, we define two different types of experiments named *Urban Low Speed tests* (uLs) and *Urban High Speed tests* (uHs) to study the influence of the speed in the performance of the VANET. In the first ones, the vehicles speed fluctuates between 20 and 30 km/h. In the *Urban*

FIGURE 7.2: Vehicles equipment used to perform VDTP real world file transfers.

High Speed tests, the vehicles speed fluctuates between 40 and 50 km/h. After the file transfers, the VDTP QoS is evaluated in terms of the number of lost packets during the downloads and the effective transmission data rate of the network during the file transfers. More information about the definition of the testbed is detailed in Toutouh and Alba (2011c).

7.2.2 Numerical Results

We present here the experimental results from two points of view. First, we study the communications carried out between the cars during the experiments in order to evaluate the VDTP service. Next, we discuss about the performance of each VDTP parameterization taken into account in this work in order to compare them with each other.

VANET Global Performance

During the experimentation, all files are transferred completely and correctly. This is mainly because the two cars (network nodes) are always following the same course (see Figure 7.1). Therefore, even though there are lost packets because of networks problems related with the distance or the existence of obstacles between the nodes, they are able to reconnect with each other before refusing the file transfer.

Table 7.1 presents the results obtained during the whole experimentation: the average number of lost packets during the transference of a file and the average effective transmission data rate performed during the downloads. The results are grouped by the VDTP parameterization used (PSO, DE, ES, GA, SA, and EXPERTS), the file type transferred (100 kBytes, 500 kBytes, 1 MByte, 5 MBytes, and 10 MBytes), and both experiment types (uLs and uHs).

Globally, in terms of transmission data rates, the majority of the file transfers are carried out with a competitive bandwidth higher than 600 kBytes/s. As expected, we check that the communications perform better (larger data rates and smaller packet loss) when the speeds of the vehicles are lower. In Table 7.1 (last row), we can observe that, in average, during uLs there are 0.133 lost packets per file transfer with an effective data rate of 610.056 kBytes/s. In contrast, during uHs there are more packet loss (0.1533) and lower bandwidth (598.878 kBytes/s).

TABLE 7.1: Average number of lost packets per transfer and effective transmission data rates (*TDR*) in kBytes/second achieved by the different VDTP configurations during the outdoor experiments.

File Size	Configuration	*Urban Low Speed* test		*Urban High Speed* test	
		Lost packets	*TDR*	Lost packets	*TDR*
100 kBytes	PSO	0.0	409.071	0.0	398.852
	DE	0.0	436.008	0.0	**476.131**
	ES	0.0	423.882	0.0	423.320
	GA	0.1	445.342	0.0	414.768
	SA	0.0	**493.717**	0.0	402.369
	EXPERTS	0.0	456.934	0.0	429.721
	AVERAGE	0.016	444.159	0.0	424.193
500 kBytes	PSO	0.0	631.454	0.0	654.180
	DE	0.0	**697.256**	0.0	645.269
	ES	0.2	470.885	0.0	651.626
	GA	0.0	677.265	0.0	677.705
	SA	0.0	646.538	0.0	**693.088**
	EXPERTS	0.0	664.802	0.0	641.828
	AVERAGE	0.033	631.367	0.0	660.616
1 MByte	PSO	0.0	717.596	0.0	696.155
	DE	0.2	582.832	0.1	628.457
	ES	0.0	**723.764**	0.0	684.709
	GA	0.2	679.841	0.0	**715.286**
	SA	0.0	704.318	0.1	708.165
	EXPERTS	0.0	691.338	0.0	678.908
	AVERAGE	0.066	683.281	0.033	676.232
5 MBytes	PSO	0.0	698.371	0.2	**659.464**
	DE	0.0	**740.345**	0.1	623.026
	ES	0.2	668.905	0.2	643.788
	GA	0.0	688.538	0.3	620.872
	SA	0.2	563.724	0.2	656.115
	EXPERTS	0.2	600.207	0.2	574.677
	AVERAGE	0.1	660.015	0.2	629.657
10 MBytes	PSO	0.4	643.576	0.4	**621.259**
	DE	0.4	**690.145**	0.4	609.239
	ES	0.2	618.988	0.5	604.702
	GA	0.6	650.997	0.6	613.021
	SA	0.6	578.297	0.7	581.800
	EXPERTS	0.5	606.760	0.6	532.147
	AVERAGE	0.45	631.460	0.533	593.695
GLOBAL		0.133	610.056	0.153	598.878

We have studied the influence of the file size in the effective transmission data rates of the downloads. In average, the 1 MByte files are transferred with the highest bandwidth (see Figure 7.3). These files are transferred with an effective data rate of 683.281 kBytes/s and of 676.232 kBytes/s during the uLs and uHs, respectively (see Table 7.1). This is because the VDTP protocol configurations used in this study are optimized to exchange data stored in 1 MByte files (see Chapter 4). The transfers of files of 500 kBytes and 5 MBytes are exchanged with data rates between 629.657 kBytes/s and 660.616 kBytes/s. The smallest data rates are obtained when the 100 kBytes files are transferred. This is because of the impact of the handshaking process of VDTP (FIRP and FIRQ packets exchange), that is greater for the smaller files. The lesson learned is that, as the nodes speed, the sizes of the files to transfer determine the performance of the VANET.

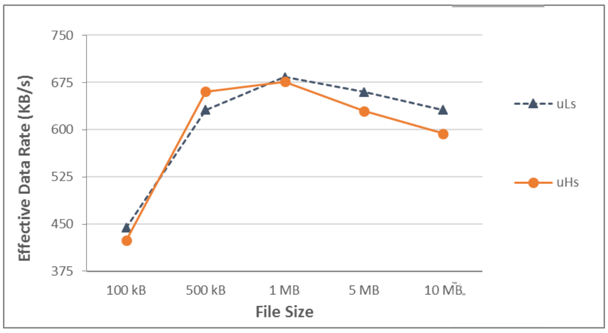


FIGURE 7.3: Average effective transmission data rates achieved during the outdoor experiments of transferring files using VDTP.

VDTP Configuration Comparison

We compare here the performance of the different VDTP configurations in terms of effective transmission data rates (also named *goodput*). Figure 7.4 shows these results achieved during the experiments. However, checking the information provided by this figure and by Table 7.1, it is not evident how to provide any global conclusion about which configuration performs the best. Thus, we have decided to carry out a statistical analysis of the results.

As the distribution of the results (75 goodput values for each VDTP parameterization) are not normally distributed, we perform non-parametric tests. Specifically, we apply the Friedman Ranking statistical test to the distributions the results of both tests, the *Urban Low Speed test* and the *Urban High Speed test*, and to the whole experimentation (named *Urban*). Thus, we are able to compare the VDTP configurations in different contexts. The confidence level is set to 95% (*p-value*=0.05).

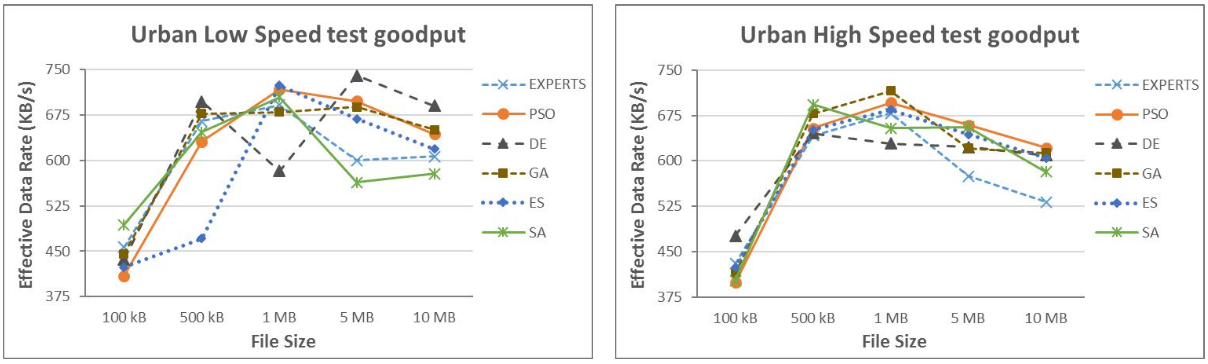


FIGURE 7.4: Effective transmission data rates (kBytes/s) achieved during the outdoor experiments of transferring files using VDTP.

TABLE 7.2: Friedman Rank test results for VDTP outdoor experiments.

Urban Low Speed test		Urban High Speed test		Urban	
Configuration	Rank	Configuration	Rank	Configuration	Rank
PSO	**4.26**	**PSO**	**4.26**	**PSO**	**4.26**
GA	3.95	SA	3.62	ES	3.60
DE	3.73	ES	3.56	GA	3.54
ES	3.64	DE	3.28	DE	3.51
SA	2.74	EXPERTS	3.16	SA	3.18
EXPERTS	2.68	GA	3.12	EXPERTS	2.92

The results of this statistical test (see Table 7.2) rank the VDTP parameterization computed by PSO as the best one during both, low and high speed tests. The average goodput of this configuration is 620.014 kBytes/s for uLs, 605.982 kBytes/s for uHs, and **612.998 kBytes/s** without taking into account the speeds. EXPERTS configuration shows the worst rank among the compared VDTP configurations for uLs and Urban, and the second worst rank for uHs. Besides, the experts configuration shows the lowest average goodput (**587.732 kBytes/s**). Therefore, this confirms *in vitro* what we observed *in silico* (by means of simulations presented in Chapter 4). It presents the PSO configuration as the most competitive one. Additionally, CARLINK experts configuration has achieved the lowest bandwidth during the simulations carried out in the experiments of that chapter. A further detailed analysis of these experiments are presented in Toutouh and Alba (2011c).

7.2.3 General Discussion on VDTP Performance Analysis

After performing the outdoor tests of VDTP file transfers, we have observed that VDTP is a competitive solution for transferring files in a VANETs. It is able to perform file transfers an average effective data rate of **603.842 kBytes/s**. The use of counters and timers of VDTP allows to complete all the file transfers, hiding the possible problems caused by link loss.

Besides this, analyzing the QoS obtained for each VDTP configuration applied in our experiments, we have observed that the automatically optimized VDTP configurations by using NC outperform the one proposed by the human experts. Specifically, the VDTP parameterization computed by PSO is the most competitive one.

7.3 Lightweight Personal Devices for VANETs

In this section, we present a set of outdoor experiments to analyze VANET communications by equipping vehicles with three different type of widespread devices (smartphones, tablets, and laptops). The idea is to prove the feasibility of VANET short range communications when using these devices. Moreover, we analyze other kinds of characteristics of these devices that are useful for VANET applications, e.g., the human machine interface (HMI) provided. Thus, we want to analyze the possibility of using such widespread lightweight commodity devices to provide ITS services to improve road transport in nowadays scenarios, where specific VANET devices are not available to most road users. The main goals of our work are: **i)** analyzing the main features that personal mobile devices provide for deploying VANETs without having to acquire new equipment for vehicles and **ii)** studying the wireless capabilities of the devices analyzed in order to discuss their use in the deployment of VANETs.

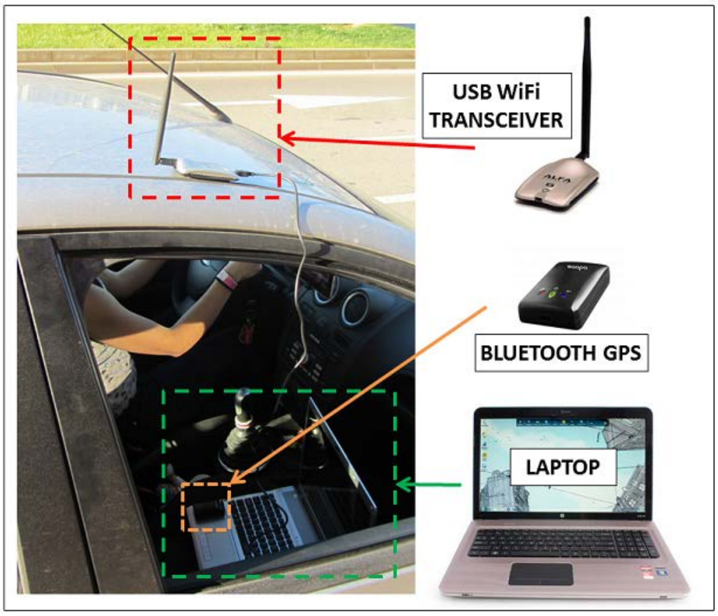
In the following, we present in more detail the three types of analyzed devices. Then, the experimental analysis is described and the results are discussed.

7.3.1 VANET Ubiquitous Devices

In our testbed, three different kinds of VANET nodes are defined depending on the type of the device used to equip the vehicles: a smartphone, a tablet or a laptop. To this end we use three devices that are readily available in the present market, as well as fairly standard in the features we are evaluating here: *Samsung Galaxy SII (Gt - I9100)* smartphones, *Samsung Galaxy Tab Gt - P7510* tablets (Samsung, 2015), and *HP Pavilion dv6-3181ss* laptops (HP, 2015) that use *ALFA AWVS051NH* USB Wi-Fi transceivers (ALFA, 2015). Figure 7.5 shows the devices used in our experimental VANET nodes.

a) Smartphone.

b) Tablet.



c) Laptop.

FIGURE 7.5: Vehicle equipment used to perform the data transfers.

The possible use of smartphones and tablets to deploy VANETs could accelerate the development of these kinds of networks, as these devices are already widely used by the population, this would in turn ease VANET's market penetration while the supply of on-board units specifically designed for vehicular communications is not widespread. In turn, these devices themselves are equipped with high processing capabilities (multi-core CPUs), many useful sensors for VANET applications (GPS, accelerometer, thermometer, compass, etc.), along with the required wireless connectivity (Bluetooth, Wi-Fi, and cellular links). Next, we present these three devices that have been used to deploy the VANET nodes of our testbed.

Smartphones/Tablets for Equipping VANET Nodes

Smartphones and tablets provide a complete set of solutions that could fulfill the requirements of a VANET node (see Figure 7.6). In order to perform V2V/V2I, V2B (cellular), and in-vehicle communications they include IEEE 802.11bgn, cellular, and Bluetooth interfaces, respectively. The Bluetooth interface may be used to interact with the on-board diagnostics (OBD) reader and with other sensors of the vehicle. The internal GPS antenna makes the vehicle's geolocation possible, which is required by most ITS services and applications.

Smartphones and tablets principally offer three different HMIs: the touch-screen, the microphone, and the speakers. The later two are really useful because drivers should not have to use their hands nor look away from the road to interact with the devices, thereby avoiding distractions that can cause dangerous situations on the road. Both devices offer several internal hardware components such as one or two cameras, a USB interface, and a set of sensors. These components can be used by different types of services or applications, e.g., the accelerometer can be used to detect a collision to inform of possible road accidents.

In terms of wireless communication capabilities, both types of devices used here integrate the same *Broadcom BCM4330* wireless chip (Broadcom, 2015), but the tablet is bigger than the smartphone, and therefore, the antenna is larger and better located in tablet devices. Finally, in our testbed, we require the vehicles during the experiments to be located, and therefore, we utilize the GPS antenna integrated in these devices.

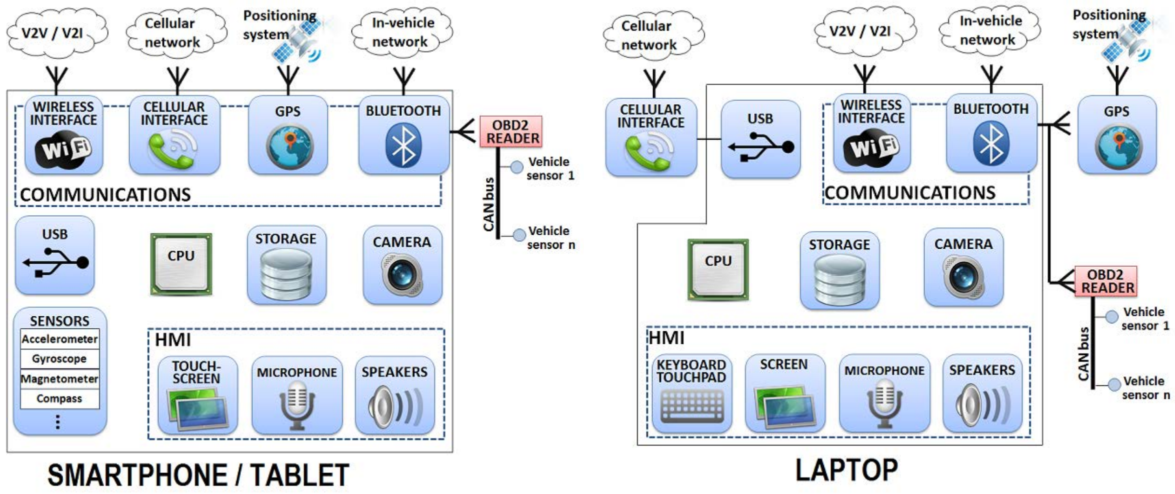


FIGURE 7.6: Main features of the analyzed devices which can be used in VANETs.

Laptops for Equipping VANET Nodes

Most laptops provide two different types of wireless interfaces: IEEE 802.11abgn and Bluetooth (see Figure 7.6). The first one can be used for V2V and V2I communications, and Bluetooth for in-vehicle communications. In order to perform V2B cellular network communications the laptop requires an external modem. In general, laptops do not include GPS antennas, so an external one may be required (see Figure 7.5.c).

In the tests carried out in this study, we evaluate the V2V communications by using a laptop as the on-board unit. The laptops use an external *ALFA AWVS051NH* USB Wi-Fi transceiver, which includes an omnidirectional gain antenna that provides 2.5 dBi for transmitting on the 2.4 GHz band and 5 dBi for transmitting on the 5 GHz band. We also use the external Bluetooth GPS antenna to locate the vehicles during the experiments (see Figure 7.5.c). In terms of HMI, laptops usually provide a larger screen than smartphones and tablets, in addition to the keyboard, the touch-pad, the microphone, and the speakers. However, the idea is to use the speakers and the microphone to interact with the system to avoid possible distractions.

For the interaction with the vehicle, as in the case of mobile personal devices, a Bluetooth OBD reader can be used. It is important to take into account that smartphones and tablets provide a set of extra sensors, but laptops do not. So, ITS applications and services can access the information provided for the internal sensors installed in the car and the external sensors connected via USB or Bluetooth interfaces. Finally, there is a drawback in using laptops as the main device for VANET nodes, which is the need for a specific space to place it in the car. This space is larger than the one needed to place smartphones or tablets.

7.3.2 VANET Testbed Definition to Evaluate Personal Devices

The experiments carried out in this study aim to analyze the feasibility of using personal portable devices to deploy VANETs. Therefore, we have designed two different types of experiments: first, the evaluation of the power of the wireless signal generated by the analyzed devices, and second, tests to characterize the QoS of V2V communications when nodes use smartphones, tablets, and laptops.

VANET Scenarios

This VANET testbed has been also defined in an urban area of Málaga covering a zone of the campus of the University of Málaga (see Figure 7.7). The two experiments described here are both carried out in this area.

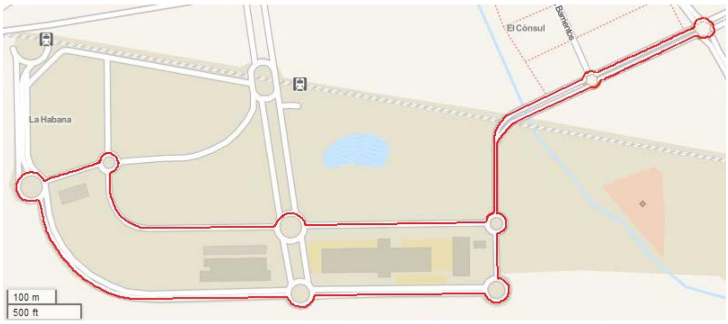


FIGURE 7.7: Urban area where the personal devices are tested as VANET nodes.

The wireless interfaces of the analyzed devices are configured with the widely used IEEE 802.11g mode that operates on the 2.4 GHz band and provides interoperability with IEEE 802.11b standard, and therefore, these devices can exchange information with most end user wireless devices. We also studied the communications between laptops when applying the IEEE 802.11a standard that operates on the 5 GHz band. This standard is also included because it operates on the closest band to the one used by IEEE 802.11p (ETSI, 2010).

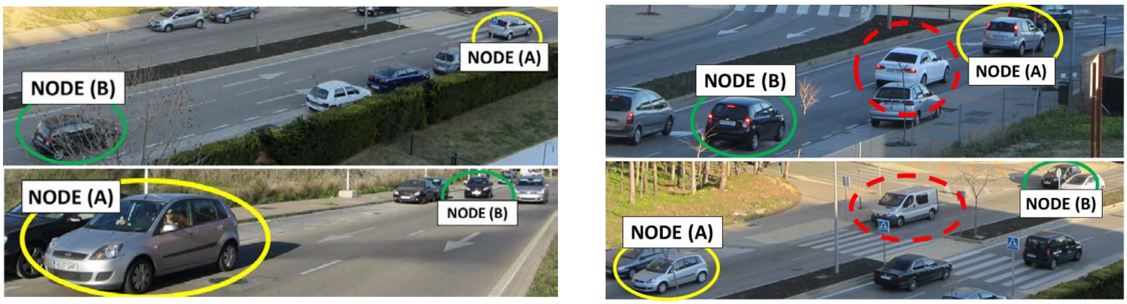
Description of the Tests

The evaluation of the **power of the wireless signal** generated by the devices studied is carried out by measuring the signal strength at different points located at distances between 0 and 150 meters during 15 seconds. In this case the cars are stopped at a given point and we use another portable device to evaluate the power of the signal. As the cars are stopped we name these experiments *static experiments*. These experiments involve the signal strength evaluation of each device in 21 different points. The points are separated from each other by 7.5 m (see Figure 7.9.a). The distance of each point i ($distance_i$) with the evaluated device is defined by Equation 7.1.

$$distance_i = i \times 7.5\ m\ \forall i \in \{0, 1, ..., 20\} \tag{7.1}$$

The transmission power is important in wireless communications since it determines different aspects of the performance of the node in the network. Increasing the transmission power, not only increases the effective coverage and reduces the attenuation rate, but it also generates a greater amount of noise (Whitehouse et al., 2007). Thus, the most efficient transmission power is difficult to determine because of the combination of these positive and negative effects. However, empirical studies have demonstrated from an application point of view that the greater the signal strength received the better the PDR (Zhao and Govindan, 2003).

The study of the **feasibility of V2V communications** is done by defining a VANET comprising two cars that exchange data with each other. These cars move along a regular road of the area that defines a circuit of 3.8 km (see Figure 7.7). As the road is open to traffic, other vehicles are able to circulate, and therefore, our vehicles are affected as in a real life journey (neither constant speeds nor fixed distances between each other). In fact, other vehicles on the road could come in between the VANET nodes, thus affecting the wireless signal propagation (see Figure 7.8.b). During the experiments, the speed of the vehicles is between 15 and 50 km/h. These experiments are called *dynamic experiments*.

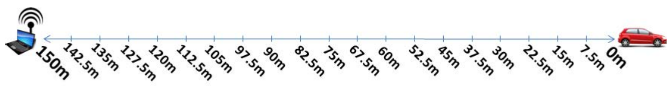


a) Cases with no vehicle in-between our two moving cars.

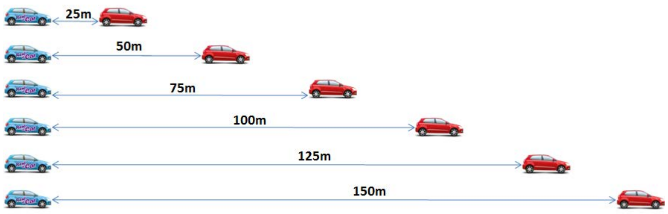
b) Cases of vehicles (in dotted circles) coming between our two moving cars.

FIGURE 7.8: Nodes during the real-wold V2V experiments.

In order to study how the distance between the VANET nodes also influences the performance of the communications, we have carried out different tests by modifying the distance between the vehicles over the journey in the *dynamic experiments*. As it was very difficult to maintain the same distances between the nodes as the ones defined for the experiments of the signal strength analysis because of the speed variations of the cars, we have used longer distances (distances multiple of 25 m). Thus, as shown in Figure 7.9.b, we have defined different experiments with the two vehicles separated by six different distances grouped in *medium distance* (25, 50, and 75 m) and *long distance* (100, 125, and 150 m).



a) Distances at which the signal strength of the devices was measured.

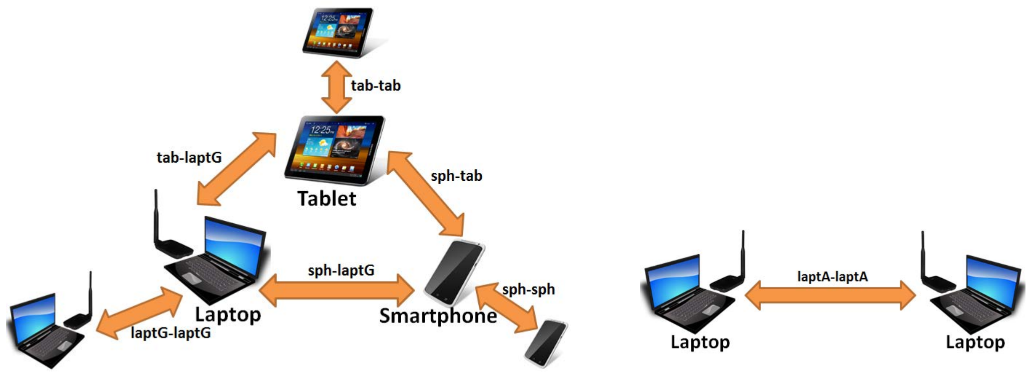


b) Distances between the cars during the V2V QoS analysis.

FIGURE 7.9: Distances between the moving nodes during the real world experimentation.

Seven kinds of VANET communication schemes are analyzed in the *dynamic experiments*. These communication schemes are distinguished by the devices connected to exchange the information and the IEEE 802.11 standard. They are named using the following format: *<device1>-<device2>*. Note that for the laptops we have included a capital letter to specify the PHY/MAC standard used, i.e., "A" for IEEE 802.11a and "G" for IEEE 802.11g. The smartphones and the tablets always communicate using IEEE 802.11g, therefore, we have not used any letter to specify the standard. Thus the communication schemes analyzed here are: smartphone-smartphone (*sph-sph*), smartphone-tablet (*sph-tab*), smartphone-laptopG (*sph-laptG*), tablet-tablet (*tab-tab*), tablet-laptopG (*tab-laptG*), laptopG-laptopG (*laptG-laptG*), and laptopA-laptopA (*laptA-laptA*). Figure 7.10 summarizes these seven communication schemes.

We have carried out 42 different tests for the *dynamic experiments*. Each test is defined according to one of the seven aforementioned communication schemes and one of the six distances between nodes (see Section 7.3.2). These tests are named according to the communication scheme and the distance: <*communication scheme*>-<*distance*>. For instance, *sph-sph-25m* refers to the test in which both nodes are equipped with smartphones and are separated by 25 m while they are moving, and *tab-laptG-125m* represents the test in which one of the vehicles is equipped with a tablet and the other with a laptop (IEEE 802.11g) separated by 125 m.



a) Communication schemes that use IEEE 802.11g (2.4 GHz).

b) Communication scheme that uses IEEE 802.11a (5 GHz).

FIGURE 7.10: Representation of the communications analyzed in the real world V2V experiments.

The V2V transferring data tests consist in exchanging data streams from the source node to the destination node of the VANET while the cars are moving. The information exchange is done by using ICMP (Internet Control Message Protocol) packets (Tanenbaum, 2002) that encapsulate the bytes of data to be sent. We have selected this protocol as the use case protocol because we do not intend to test any specific type of application. In order to study the effects of the size of data packets on the performance of the VANET communications, we have performed data transfers by exchanging five kinds of data packets for each V2V test, that are defined by their size (32, 64, 128, 256, and 512 bytes of data). For each one of the five types of data packets, each node transfers to the other streams of 100 independent packets each.

For more detailed information about the VANET testbed see (Toutouh and Alba, 2016). Now let us to present the results in terms of transmission power and QoS for the experimental analysis performed in this work.

7.3.3 VANET Nodes Transmission Power Analysis

In the *static experiments*, the wireless transmission power of each device is evaluated in terms of the *received signal strength indicator* (RSSI) in a given location. RSSI reflects relative received signal strength in a wireless environment, in arbitrary units. Specifically, RSSI is an indication of the power level observed by a radio hardware while receiving a data frame. Remember that the evaluated RSSI includes the power from adjacent channel interference, thermal noise, etc. that could affect the signal received. In order to measure the RSSI we use dBm (also known as dBmW), which is an abbreviation for the power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW).

The experimental results are summarized in Figure 7.11, which shows the average signal strength sensed at a given point for each studied device. The shaded area represents the lowest receiver sensitivities for which a receiver can correctly decode frames for most of IEEE 802.11g-based wireless devices (signal strengths between -90 dBm and -80 dBm).

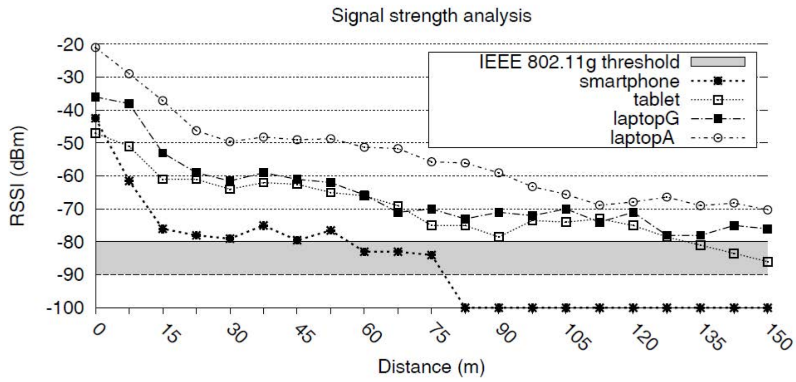


FIGURE 7.11: Signal strength results the outdoor V2V experiments.

As a first conclusion, the RSSI decreases with the distance between the node (device) that generates the signal and the measurement point (see Figure 7.11). At the same time, we observe that, even if we have measured the signal strength for some time to limit the negative effects over the signal propagation in real outdoor scenarios, the values still show some irregular behavior. Our explanation for this phenomenon is that the RSSI includes the received power of the interference, thermal noise, etc. that affect the possible regular behavior of such metric.

As expected, the **smartphone** is the least competitive device in terms of RSSI results. The signal strength values for this device are between -90 dBm and -80 dBm from 60 m to 75 m, which means that its performance in transmitting information could suffer from a degradation at these distances. After 75 m, the RSSI is lower than -90 dBm, and therefore, smartphones barely have the capacity to exchange data streams at greater distances.

The strength of the signal produced by the **tablet** is higher than -70 dBm for distances lower than 67.5 m. After that, from this point until 127.5 m, the strength of the signal decreases but it maintains values greater than -80 dBm. Thus, tablets may offer competitive communication performances at distances up to 127.5 m. Finally, when the distance is greater than **127.5 m** the signal strength is lower than -80 dBm, but always higher than -90 dBm, which means that the tablet signal strength is over the lowest sensitivity threshold of IEEE 802.11g wireless based communications throughout our experiments.

The best results in terms of signal strength are provided by the **laptop** when using both PHY/MAC standards, IEEE 802.11a and IEEE 802.11g. On the one hand, when IEEE 802.11a is used, the received signal strength is significantly higher than in the other devices. The main reason is that the antenna gain is higher on the 5 GHz band than on the 2.4 GHz. In this case, the RSSI is lower than -60 dBm just when the distance is longer than 90 m and the lowest measured signal strength is -70.29 dBm (see Figure 7.11). On the other hand, when the radio used in the laptop is configured with IEEE 802.11g, the RSSI results when the distance is lower than 67.5 m are surprisingly close to the ones obtained by the tablet. From this point until the furthest one the **signal strength is always higher than -80 dBm**.

Therefore, the best results in terms of transmission power are achieved by the laptop when transmitting on the 5 GHz band. The second best ones are achieved when the same device uses the 2.4 GHz band. The third best RSSI is shown by the tablet, allowing interesting competitive results compared to the laptop for the complete experimentation (RSSI higher than -90 dBm). The lowest signal strength results are presented by the smartphone: so after 75 m the experimental results are lower than -90 dBm, offering an undesirable behavior. Applications needing more than 75 m would probably not rely on smartphone's Wi-Fi, with the important exception of V2I (e.g., exchanging information with Wi-Fi spots). We must also mention that, even if these results were expected, we are here quantifying the distance ranges and communication power of regular smartphones, something difficult to find in the related literature.

Finally, note that, although the signal strength decreases with the distance for all devices and frequency bands, the reduction is smoother and the signal strength is more stable when the radio used the 5 GHz band. This happens because the 2.4 GHz frequency band is way more crowded than the 5 GHz one, and therefore, the devices on the 2.4 GHz suffer much more interference than those on the 5 GHz.

7.3.4 VANET Communication Feasibility Experimental Results

In this section, we present the experimental results, analyzing the exchange of data between the two moving vehicles by evaluating E2ED, PDR, and TDR (these metrics have been already defined in Section 4.1.2). Table B.18 in Appendix B, illustrates the whole results. Table 7.3 summarizes the results of the entire experiment by showing the average and the relative standard deviation of the three metrics studied. Thus, it is easier to conduct a comprehensive study of the performance of VANET communications by using smartphones, tablets, and laptops. The results are grouped by distances between the nodes (*medium distance* and *long distance* experiments). It also presents the *global average* results for the complete *dynamic experiments* set.

TABLE 7.3: Average and relative standard deviation E2ED, PDR, and TDR results of the personal devices outdoor testbed grouped by the distance between vehicles.

Connection	E2ED (ms)		PDR (%)		TDR (KB/s)	
type	Avg.	Stdev.	Avg.	Stdev.	Avg.	Stdev.
			medium distance (from 25m to 75m)			
sph-sph	140.68	181.43%	83.11	32.21%	1.99	96.02%
sph-tab	30.47	33.54%	81.33	27.22%	7.46	90.67%
sph-laptG	13.06	51.68%	**98.45**	2.82%	18.26	80.90%
tab-tab	56.43	52.82%	94.89	7.14%	3.91	69.69%
tab-laptG	21.78	99.86%	92.67	11.63%	19.05	116.86%
laptG-laptG	**1.68**	29.43%	**98.45**	3.81%	**128.53**	69.87%
laptA-laptA	16.10	156.96%	40.76	18.29%	13.80	71.39%
			long distance (from 100m to 150m)			
sph-sph	-	-	0.00	0.00	0.00	0.00
sph-tab	-	-	0.00	0.00	0.00	0.00
sph-laptG	13.97	41.18%	86.45	14.54%	16.94	83.88%
tab-tab	144.19	98.85%	43.67	68.73%	1.52	75.95%
tab-laptG	64.71	87.39%	91.67	8.44%	9.11	146.81%
laptG-laptG	**3.06**	63.29%	**94.45**	8.08%	**72.86**	67.42%
laptA-laptA	156.01	63.80%	8.67	157.57%	0.44	192.93%
			global average (from 25m to 150m)			
sph-sph	101.33	251.88%	41.56	111.12%	0.99	167.79%
sph-tab	15.70	65.10%	40.67	108.52%	3.73	161.92%
sph-laptG	14.48	42.65%	92.45	11.71%	17.60	81.01%
tab-tab	101.60	110.45%	63.99	59.38%	2.51	99.59%
tab-laptG	47.59	101.79%	92.17	10.02%	14.08	133.09%
laptG-laptG	**2.51**	63.59%	**96.45**	6.48%	**100.71**	76.01%
laptA-laptA	38.42	259.38%	24.71	79.21%	7.12	135.65 %

According to the results in Table 7.3, the communications in the testbed scenario in which both nodes are equipped with **tablets** (*tab-tab* tests) require longer E2ED than the scenario in which one node uses a **smartphone** and the other uses a tablet (*sph-tab* tests). This is not the expected behavior because the smartphone signal strength is lower than the one of the tablet (see Section 7.3.3), and therefore, our smartphone should show worse wireless communication capabilities than our tablets. The same occurs when we compare the E2ED of the *tab-laptG* and the *sph-laptG* tests. This can be explained by the road traffic density growth during the experiments with tablets (*tab-tab* and *tab-laptG*), as Figure 7.8.b illustrates. Thus, the data transfers suffer from the existence of obstacles between the nodes.

Taking into account just the experiments carried out by using two **laptops**, we can observe that the average E2ED during the *laptA-laptA* tests is longer than ten times the E2ED during the *laptG-laptG* tests. This difference is significantly greater if we take into account the average E2ED during the *long distance* tests (E2ED *laptA-laptA*=156.01 ms and *laptG-laptG*=3.06 ms). This is principally because the performance of the network is more likely to be negatively affected by real world obstacles when it uses a higher frequency (Doefexi et al., 2003). Thus, the IEEE 802.11g standard is more promising than the IEEE 802.11a to perform competitive vehicular communications (**average E2ED *laptG-laptG*=2.51 ms**), while the market does not broadly assimilate the use of IEEE 802.11p.

The data transfers performance suffers from variability of the communications provoked by the obstacles (e.g., other vehicles). This can be observed in the standard deviation results in Table 7.3. As the road traffic increased during the experiments involving the tablets, the *tab-tab* and the *tab-laptG* transmissions presented larger deviation values than the *sph-tab* and the *sph-laptG* ones, respectively. The largest deviation value, which means the lowest robustness, is obtained by the laptop communicating on the 5 GHz band (*laptA-laptA*).

Figures 7.12 and 7.13 summarize the experimental results in terms of PDR and TDR, respectively, grouped by the distances between the VANET nodes. Note that, in both figures, some bars that represent the results of *sph-sph*, *sph-tab*, and *laptA-laptA* tests do not appear because there has been no data information exchange due to the complete loss of communication, as it can be seen in Table 7.3.

Analyzing the quantity of the successfully **delivered data packets (PDR)**, we observe two clearly differentiated behaviors. On the one hand, all the VANET communications studied between devices that used the IEEE 802.11g standard (smartphones, tablets, and laptops) present similar and competitive results (**PDR above 80%**) for the scenarios in which the vehicles are separated by *medium distance* (see Table 7.3). However, only communications that involve laptops maintain such a high performance (PDR above 85%) when the distance passes beyond 100 m (*long distance*). On the other hand, when the vehicles exchanged data by using laptops configured with the IEEE 802.11a standard, the communications showed a limited performance since the PDR results were always below 50%. However, these nodes were not able to perform any data exchange when they were separated for distances greater than 100 m.

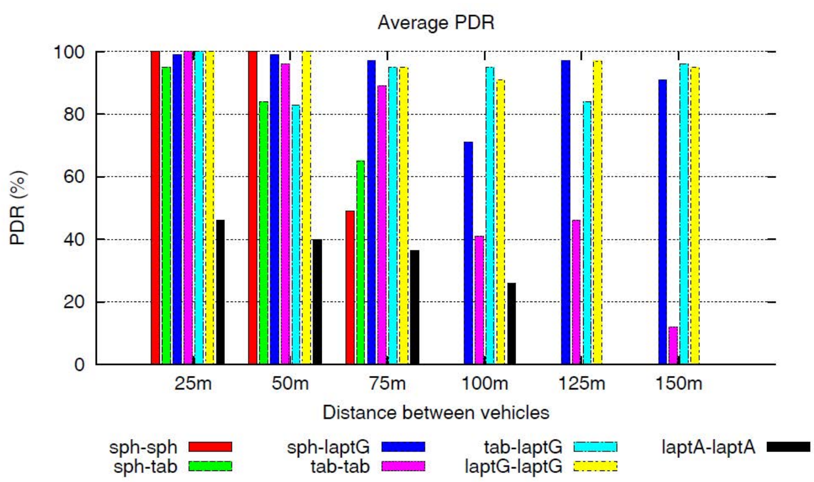


FIGURE 7.12: Average PDR results of outdoor testbed of V2V personal devices communication.

Laptops transferred the highest amount of data between each other transmitting on the 2.4 GHz band (see Figure 7.12). The average PDR for the *laptG-laptG* communications is **98.45%** in *medium distance* and **94.45%** in *long distance*. In any case, the global average PDR is higher than 90% for all tests in which at least one of the communication nodes is equipped with a laptop and is using the IEEE 802.11g standard (see Table 7.3).

This is in sharp contrast to the results achieved by the same devices (laptops) when they communicate using IEEE 802.11a. The amount of delivered data are 40.76% and 8.67% for *medium distance* and *long distance*, respectively. In turn, the effective transmission range is shorter for the IEEE 802.11a communications than for the IEEE 802.11g ones. The main reason for this is that both standards use the same modulation (OFDM) and IEEE 802.11a transmits on higher frequencies (5 GHz over 2.4 GHz), reducing its communication capabilities against IEEE 802.11g (Al-Khusaibi et al., 2006; Paul et al., 2011).

In terms of **transmission data rates (TDR)**, there are considerable differences between the *laptG-laptG* communications and the other VANET communications studied here (see Table 7.3). For this reason, Figure 7.13 is shown in logarithmic scale.

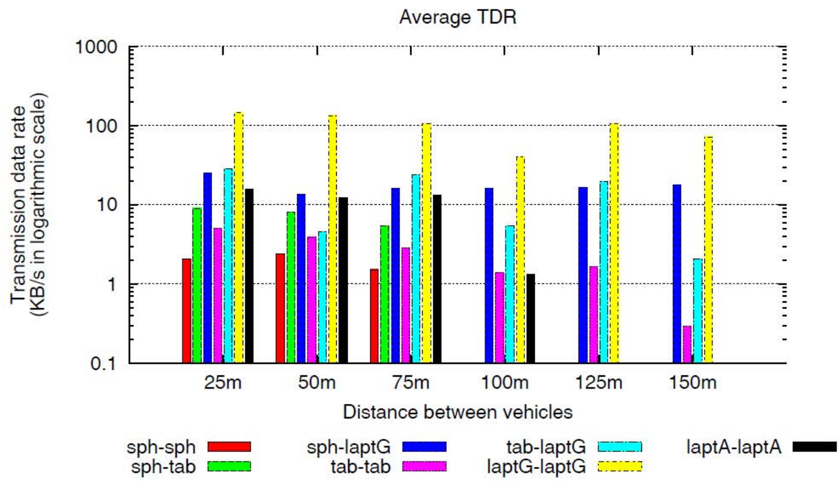


FIGURE 7.13: Average TDR results of outdoor testbed of V2V personal devices communication.

Smartphones have the least competitive results in terms of TDR. In the *sph-sph* and *sph-tab* experiments, the nodes are able to exchange data in just *medium distance* scenarios by achieving average data rates of 1.99 KB/s and 7.46 KB/s, respectively. Although it seems a poor performance, the TDRs achieved for distances between 25 and 75 m are enough to deploy VANET applications for exchanging lightweight warning messages and traffic information with vehicles nearby and with the authorities via V2I.

Tablets offer better TDR results than smartphones since *tab-tab* experiments achieve higher rates (its global average TDR 2.51 KB/s) and communicate with nodes at greater distances (up to 125 m). However, when one of the nodes is equipped with a laptop this does not hold true. The *sph-laptG* tests obtained more competitive TDR results than *tab-laptG* ones (global average TDRs, *sph-laptG*=17.60 KB/s and *tab-laptG*=14.08 KB/s). This is due to the increase in road traffic during the test that involved vehicles equipped with tablets, which negatively affected the communications.

Taking into account the experiments carried out by transferring data by using laptops configured with the IEEE 802.11a standard, the performance is significantly different for both *medium distance* and *long distance*. When the distance is shorter or equal to 75 m, these communications achieved a competitive transmission data rate (average TDR *laptA-laptA*=13.80 KB/s). However, the TDR drops to 0.44 KB/s for *long distance* tests.

As in the previously analyzed metrics, in the analysis of the data rates, the best results are achieved when the two VANET nodes use laptops using IEEE 802.11g to exchange data between each other. In the *medium distance* experiments the average ***laptG-laptG* TDR is 128.53 KB/s** and in the *long distance* ones the average TDR is **72.86 KB/s** (see Table 7.3). These competitive transmission data rates may allow the exchange of multimedia information between the nodes, such as voice messages and videos. It is important to remark that this high TDR is achieved in distances up to 150 m.

Broadly speaking, the feasibility of the communications carried out on the 2.4 GHz, vehicles equipped with laptops offer a practical solution for deploying VANETs right away, because this solution allows communications at distances greater than 150 m with the largest PDR and TDR and the lowest E2ED. This is not an unexpected behavior because the laptops are equipped with the highest gain antenna that is placed outside the car (see Figure 7.5.c). Analyzing the personal portable devices, both present competitive QoS and so could deploy useful VANETs with applications that do not require large transmission data rates. At the same time, tablets provided an effective communication coverage over 125 m, and smartphones a coverage up to 75 m. These relatively low performance results dramatically improve when devices with powerful wireless interfaces, such as laptops, are included in the VANET communication loop.

Note that, even the results obtained when the radios used the 5 GHz band are less competitive, the IEEE 802.11p communications are still considered as a promising technology for vehicular environments. This standard provides several improvements that make it more robust than IEEE 802.11a. For instance, the reduction of the bandwidth of the channels from 20 MHz of IEEE 802.11a to 10 MHz of IEEE 802.11p, which duplicates the transmission time for a specific data symbol, allowing the receiver to better cope with the characteristics of the radio channel in vehicular environments (Lin et al., 2012).

Finally, the importance of these outdoor testbeds is notable, because some of our experiments have suffered from the existence of real world obstacles producing a decrease in the performance of the wireless communications, leading to far from ideal results. A further detailed analysis of these experiments are presented in Toutouh and Alba (2016).

7.3.5 General Discussion on V2V by Using Lightweight Devices

Widely available smartphones and tablets provide a set of facilities, which are required by VANET applications. Thus, they can be used to deploy VANETs even if specific on-board units are not available to most road users. This could accelerate an early development of vehicular networks to provide useful ITS services.

In the light of the real-wold communications experimental analysis by using such devices, we can conclude that: smartphones allow useful information to be exchanged with nearby nodes in urban areas; tablets improve the smartphones' communication capabilities and could be used to exchange more information, also on highways; finally, laptops are able to exchange multimedia information (audio and video) with the highest data rates and with any kind of communication partner node at distances up to 150 m.

7.4 Conclusions

In this chapter, we have analyzed vehicular communications by defining two real world experiments carried out in open roads of the city of Málaga. They have been defined for two different main purposes: **1)** confirming *in vitro* what we observed *in silico* about the use of optimized VANET protocols in Chapter 4 and **2)** evaluating the possibility of using the widely available lightweight devices in VANET communications.

For the first purpose, we have focused on the VDTP file transfer protocol. Thus, we have analyzed the file transfers when configuring the protocol with the human experts parameterization and with the ones obtained by using NC. The human experts configuration performed the worst in exchanging files under the same conditions. The **best results** have been obtained when using the VDTP **configuration computed by PSO**. Therefore, the experimental results confirmed the QoS improvements of the optimized VDTP experienced by simulations.

When the use of portable devices to deploy VANETs have been analyzed (instead of specific OBUs), we have observed that smartphones and tablets provide a set of facilities which can be directly utilized for VANET applications (e.g., GPS antenna). According to the VANET communications results, we have observed that **smartphones** can be used to exchange information with nodes located **up to 75 m** of distance. Nodes equipped with **tablets** provides a higher effective coverage **over 125 m**. **Laptops** equipped with a IEEE 802.11g Wi-Fi transceiver are able to exchange data with nodes at distances **greater than 150 m**, with the highest transmission data rates (**higher than 100 kBytes/s**). Besides this, when a laptop is included in the VANET communication loop the network dramatically improve the performance.

It should be taken into account that the performance of VANET communications has not always followed the regular expected behavior. This can be explained by the variability of the road traffic density (vehicles act as obstacles for the signal propagation) and by the existence of a number of other wireless networks that interfered with the signal of our VANET. This kind of behavior, which influences the radio signal propagation, and therefore the communications QoS, is quite difficult to accurately represent and evaluate in simulators or emulators. Therefore, performing real world tests is strongly recommended to evaluate vehicular networks, even if it is difficult to master when conducting such experiments.

Part III: Conclusions and Future Work

The most important reason for going from one place to another is to see what is in between, and they took great pleasure in doing just that.

Norton Juster

Chapter 8

Conclusions and Future Work

FROM the early days of humanity, nature has been an unlimited source of inspiration for the design of useful solutions, ranging from the construction of simple artifacts to the definition of complex industrial processes. In this thesis we have analyzed a set of nature inspired algorithms to address vehicular networks optimization problems (the off-line and on-line optimization of protocols and the smart design of RSUs infrastructure). These algorithms have presented a competitive performance in finding accurate solutions that improve the VANET behavior. Additionally, we have carried out VANET pilots to evaluate these communications in real world environments. This chapter contains a global review of this PhD thesis and regroups the main conclusions drawn for the whole research work. The thesis objectives and main contributions are discussed now in view of the results. To end this chapter, the future lines of research that can be pursued are briefly sketched and discussed.

8.1 Conclusions

This thesis dissertation has tackled the resolution of complex optimization problems in the domain of vehicular networks. Since this is a young field we can find in it a new and inspiring set of possibilities for research and industry, since the efficient deployment of VANETs may change modern society by enhancing one of its main issues: the road traffic. But also, VANETs brought new problems and harsh constraints which must be addressed. Considering these factors, we have analyzed the application of modern NC to deal with them.

We have first reviewed the main features of VANETs, including the basic aspects of the communication technologies that are involved in vehicular communications (i.e., architecture, communication domains, and radio access technologies), and the most salient proposed applications. We have listed the main differences between VANETs and other types of MANETs. We have summarized their main open issues that presently receive more attention from research community (with or without an optimization component).

The next step has been to review NC focusing on its strength in solving complex optimization problems. We have used different types of NC techniques (EAs, swarm intelligence, and SA) and have used different sorts of formulation problems (mono-objective and multi-objective). Then, from the main open issues mentioned in VANETs, we have selected some that successfully can be treated as optimization problems. These are categorized in two main groups: **1)** the ones regarding to data dissemination (i.e., file transfer, routing, and broadcasting), addressed as on-line and off-line optimization problems, and **2)** the one which deals with the optimal design of a roadside infrastructure.

It should be taken into account that our approaches have not been applied over abstract academic definitions, instead we have put stress onto defining realistic problem models and instances by modeling real world wireless communication and vehicular environments (based on Málaga). We describe these problems and the main results shortly below.

Regarding **off-line optimization of data dissemination**, we applied an automatic mechanism based on coupling NC and a realistic VANET simulation in order to compute optimized configurations of VANET protocols. We analyzed this problem from different points of view:

- The ***file transfer optimization (FTC)*** problem consists in computing configurations of a file transfer protocol in order to maximize the effective transmission data rate and minimize the amount of lost data. The effective transmission data rated of the computed configurations was **more competitive than the standard configuration** of the protocol. Therefore, VANET nodes may transfer larger files (e.g., multimedia) and show lower error rates when using optimized protocols.

- In the ***QoS routing optimization***, the idea is to find accurate routing protocols parameterizations that maximize the QoS, while the wireless medium is efficiently used. We optimized ***OLSR QoS*** by defining a mono-objective formulation taking into account PDR, E2ED, and NRL. The resulting configurations showed **high PDRs (>84%)**, **reducing drastically NRL** regarding the standard one. Besides this, we decided to explore the possibility of applying a multi-objective formulation in order to find a set of solutions that maximize the PDR and minimize the E2ED of AODV at the same time (***AODV MO-QoS***). The computed protocol configurations **outperformed the state-of-the-art** ones in terms of PDR while keeping the E2ED in the threshold of proper operation. Therefore, our optimized routing protocols improve both the reliability of the communications (high PDRs) and the VANET scalability by reducing the overhead (NRL), while keeping low communication delays.

- As VANETs may include nodes with energy restrictions (e.g., sensors fed with solar panels or with small batteries), a similar idea has been also analyzed in computing power-aware routing protocols. Thus, we optimized the ***energy-efficiency of OLSR*** while keeping its PDR above a given quality threshold. The energy-efficient configurations achieved **energy consumption reductions up to 40%**, improving the results of other approaches presented in the literature. The use of this power-aware protocol may facilitate the future use of nodes with energy consumption constraints.

Let us now summarize our findings concerning the NC methods utilized to address these off-line optimization problems. We initially started evaluating a set of different canonical algorithms (PSO, DE, GA, ES, and SA) because we did not find other analogous studies in the related literature. This way, a set of preliminary results was provided in order to allow future comparisons with other optimization techniques. Thus, we analyzed these algorithms in solving FTC and OLSR QoS. Globally, **PSO presented very competitive and robust results** when computing accurate parameterizations.

After this analysis, we detected that the final solutions computed by the NC algorithms suffered from the need of using a reduced number of fitness function evaluations during the optimization process due to the high computation costs of performing VANET simulations. Therefore, we decided to apply **paralleled algorithms** (i.e., pGA, pPSO, pNSGA-II, and pSMPSO) by using a master-slave model in order to distribute the fitness function evaluations through different processing elements (CPUs). This strategy provided a high **time inefficiency, which was in average between 80% and 90%**, allowing the algorithms to compute protocol configurations better than sequential techniques. Additionally, we designed problem specific operators which provided better results than the canonical ones.

Finally, as we observed that the configurations computed by applying NC provided different QoS depending on the scenarios (e.g., number of vehicles or type of applications), we decided to explore the possibility of applying a multi-objective formulation in order to compute a set of accurate configurations that **maximize the PDR and minimize the E2ED at the same time**, if possible. Thus, a given obtained optimized configuration might be applied depending on the network status or the VANET application. The results showed that the idea behind this methodology is sound. It should be noticed that the methodology presented here in off-line optimization of VANET protocols can be tailored to successfully optimize any other protocol, not only the ones evaluated here.

After this long series of off-line optimization problems (targeted to a better design and use of the VANET), we got to a not less challenging domain: real time optimization of the communications while the VANET is in operation. For this, the **on-line beacon broadcasting optimization** problem in CVS applications was addressed by tacking into account two metrics: the channel occupancy and the fairness (balance in using the channel). Thus, a set of greedy dynamic broadcasting methods were devised: two ***Self FREEDY*** algorithms, which are based on isolated monitored information, and four ***Swarm FREEDY*** algorithms, which combine monitored information with data received from the neighbor nodes, being Swarm FREEDY the most competitive ones. Self FREEDY outperformed other broadcasting base methods (Aloha and CSMA), **Swarm FREEDY were the most competitive** ones. Therefore, sharing information between the VANET nodes has shown to improve the performance of the congestion control algorithms. Though Swarm FREEDY provided competitive results, we advice they are just a first step to design other new Nature inspired CVS applications based on swarm intelligence (e.g., ACO or ABC). Another distributed greedy algorithm was proposed to address the ***RAT problem***, in which the nodes have to dynamically select which radio technology to utilize (IEEE 802.11p or LTE), however it is not presented in this manuscript because of length constraints. Further details can be found in Mir et al. (2015).

In ***RSU-DP***, in contrast to the previous studies in the literature that applied mono-objective optimization methods, we presented here a new explicit multi-objective formulation. A pEA was applied to solve a realistic problem instance based on actual information of Málaga city. The pEA significantly outperformed other two state-of-the-art greedy heuristic algorithms. In average, these improvements meant **19.0% of the cost** and **5.7% of the QoS**. In addition, we obtained a set of different solutions that present different trade-offs between the cost and the QoS. This facilitates the decision making of VANET designers, who can chose the one that best fits their specific needs and constraints.

In this thesis we have not just analyzed optimization problems over realistic instances, but in addition we performed **real world outdoor tests** in order to confirm the results obtained by using simulations. In this sense, we have defined a testbed in order to perform file transfers between cars by using **different versions of VDTP**: the optimized ones by NC and the standard one. The results showed that the automatically optimized VDTP configurations outperformed the one proposed by human experts. And as it happened during the simulations, the PSO computed parameterization was the most competitive one. Besides this testbed, we have carried out another one in order to analyze the **feasibility of using personal devices to perform V2V communications**. During the experimentation we observed that: **a)** smartphones can be used to exchange information with nodes located up to 75 m of distance, **b)** tablets provide a higher effective coverage over 125 m, **c)** and laptops are able to exchange data with nodes at distances greater than 150 m. In turn, when a laptop was included in the VANET communication loop the network dramatically improved the performance.

In short, this PhD thesis has addressed several VANET optimization problems, and NC has emerged as a satisfactory methodology for solving them. We have explored novel mono-objective and multi-objective formulations to these problems that have helped to improve the state-of-the-art results. The performance of the utilized NC techniques have been improved by applying two main strategies: paralleling the algorithms and devising new problem specific operators. The algorithms have been parallelized in order to efficiently mitigate the issue of the high computation cost (time) of evaluating the solutions, and therefore, to increase the allowed number of fitness function evaluations. Several novel evolutionary search operators (initialization, mutation, crossover, etc.) have been designed in order to improve the efficacy of the NC algorithms in dealing with such optimization problems. Moreover, new distributed dynamic greedy algorithms have been proposed in order to establish a starting point for developing more intelligent data dissemination algorithms based on swarm intelligence. Finally, real world experiments have been carried out to satisfactory prove the realism of the results obtained by laboratory simulations. Besides providing a wide plethora of useful results by themselves, we can also conclude that NC has proven to be an efficient tool to tackle optimization problems in vehicular networks.

8.2 Future Work

After this thesis work, there are still some open lines to address in the future. Regarding off-line optimization of VANET protocols, the solution evaluation during the optimization process may be carried out by using **Monte-Carlo simulation** over different VANET scenarios in parallel. Applying this idea, the algorithms may compute more robust optimized protocol configurations, as we observed that the optimized configurations performed differently depending on the VANET scenario. In addition, a **dynamic method** that allows the VANET nodes to decide which optimized protocol configuration (from a given Pareto front) should be utilized depending on the current VANET situation could be proposed in order to adapt the protocol operation to the current network constraints and requirements. Finally, **advanced multi-objective NC** algorithms may be explored in order to improve the efficiency and efficacy of the optimization process presented in this work.

Focusing on the on-line optimization of broadcasting and RAT selection, the dynamic greedy algorithms proposed in this thesis work may be improved by including more **advanced swarm intelligence** based operations, since swarm intelligence based algorithms have presented very competitive results in many other problems (e.g., MANET routing).

In the RSU-DP, the main lines of future studies are related to **extend the experimental analysis** to other geographical areas and to consider additional information (e.g., road accidents or points-of-interest) in order to model the problem by including more road traffic related information. Besides this, the problem may include **specific VANET applications** based on V2I communications (e.g., multimedia downloading for infotainment or local weather forecast broadcasting) to define communication requirements for the RSUs.

Regarding outdoor experiments, we used IEEE 802.11a (5 GHz band communication) because it operates on the closest frequency band to the one used by IEEE 802.11p, but the results were not very competitive in V2V communications. The testbed should be extended by **a) including IEEE 802.11p** devices, as soon as we are able to find them on the market, in order to confirm its feasibility for communicating VANET nodes; **b)** increasing the number of nodes; and **c)** evaluating optimized versus standard protocols.

Finally, the research work presented in this PhD thesis provides the basis for considering NC as a promising mechanism to address diverse issues in vehicular networks. Therefore, it gives the research community an innovative body of knowledge to accelerate the deployment of VANETs, a technology that will change our near future life.

APPENDICES

En tiempo de higos no hay amigos...
Mucho ib-bah!

HACH MOHAND EL ALAMIN

Appendix A

List of Publications Supporting this Thesis Dissertation

In this appendix we present the set of works that have been published while this thesis work has been developed. These publications speak for the interest, validity, and impact on the scientific community and literature of the work contained in this thesis, since they have appeared in prestigious fora, and have been subject to peer review by expert researchers. It should be noted that the working hypothesis presented in this thesis dissertation was awarded with a second prize in the ***Doctoral Consortium*** **Award** 2013 given by *Asociación Española para la Inteligencia Artificial* (Spanish Association for Artificial Intelligence) (Toutouh and Alba, 2013a). We list our publications next.

ISI JCR indexed journals [5]:

J. García-Nieto, J. Toutouh, and E. Alba (2010). "Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics". In: *Engineering Applications of Artificial Intelligence* 23.5. Advances in metaheuristics for hard optimization: new trends and case studies, pp. 795–805

- Impact Factor: **2.207**
- Category:
  - AUTOMATION & CONTROL SYSTEMS: **Q2**
  - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: **Q1**
  - ENGINEERING, ELECTRICAL & ELECTRONIC: **Q1**
  - ENGINEERING, MULTIDISCIPLINARY: **Q1**

J. Toutouh, J. García-Nieto, and E. Alba (2012b). "Intelligent OLSR Routing Protocol Optimization for VANETs". In: *Vehicular Technology, IEEE Transactions on* 61.4, pp. 1884–1894

- Impact Factor: **1.978**
- Category:
  - ENGINEERING, ELECTRICAL & ELECTRONIC: **Q2**
  - TELECOMMUNICATIONS: **Q1**
  - TRANSPORTATION SCIENCE & TECHNOLOGY: **Q2**

J. Toutouh, S. Nesmachnow, and E. Alba (2013). "Fast energy-aware OLSR routing in VANETs by means of a parallel evolutionary algorithm". In: *Cluster Computing* 16.3, pp. 435–450

- Impact Factor: **1.510**
- Category:
  - COMPUTER SCIENCE, INFORMATION SYSTEMS: **Q2**
  - COMPUTER SCIENCE, THEORY & METHODS: **Q1**

J. Toutouh and E. Alba (2015c). "Parallel multi-objective metaheuristics for smart communications in vehicular networks". English. In: *Soft Computing*. In Press., pp. 1–13. URL: http://dx.doi.org/10.1007/s00500-015-1891-2

- Impact Factor: **1.271**
- Category:
  - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE **Q3**
  - COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS: **Q3**

J. Toutouh and E. Alba (2016). "Light commodity devices for building vehicular ad hoc networks: An experimental study". In: *Ad Hoc Networks* 37, Part 2, pp. 499–511

- Impact Factor: **1.530**
- Category:
  - TELECOMMUNICATIONS: **Q2**
  - COMPUTER SCIENCE, INFORMATION SYSTEMS: **Q2**

Peer review international journals [1]:

J. Toutouh and E. Alba (2015b). "Metaheuristics for energy-efficient data routing in vehicular networks". In: *International Journal of Metaheuristics* 4.1, pp. 27–56

CORE ranked intentional conferences [4]:

J. Toutouh and E. Alba (2011a). "An efficient routing protocol for green communications in vehicular ad-hoc networks". In: *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. GECCO '11. Dublin, Ireland: ACM, pp. 719–726

- CORE Score: **A**

J. Toutouh and E. Alba (2011c). "Performance analysis of optimized VANET protocols in real world tests". In: *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference*. IWCMC. Istanbul, Turkey: IEEE, pp. 1244–1249

- CORE Score: **B**

J. Toutouh and E. Alba (2012a). "Green OLSR in VANETs with differential evolution". In: *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. ACM, pp. 11–18

- CORE Score: **A**

J. Toutouh, S. Nesmachnow, and E. Alba (2012a). "Evolutionary Power-Aware Routing in VANETs using Monte-Carlo Simulation". In: *Proceedings of The 10th International Conference on High Performance Computing and Simulation (HPCS 2012)*. Madrid, Spain: IEEE Computer Society Press, pp. 119–125

- CORE Score: **B**

International conferences of *Lecture Notes in Computer Science* series [1]:

Z. H. Mir, J. Toutouh, F. Filali, and E. Alba (2015). "QoS-Aware Radio Access Technology (RAT) Selection in Hybrid Vehicular Networks". English. In: *Communication Technologies for Vehicles*. Ed. by M. Kassab et al. Vol. 9066. Lecture Notes in Computer Science. Springer International Publishing, pp. 117–128

Other peer review international conferences [8]:

E. Alba, S. Luna, and J. Toutouh (2008). “Accuracy and Efficiency in Simulating VANETs”. In: *Modelling, Computation and Optimization in Information Systems and Management Sciences, Second International Conference (MCO)*. vol. 14. Communications in Computer and Information Science. Metz, France - Luxembourg: Springer, pp. 568–578

J. Toutouh, J. García-Nieto, and E. Alba (2010b). “Optimal configuration of OLSR routing protocol for VANETs by means of Differential Evolution”. In: *3rd International Conference on Metaheuristics and Nature Inspired Computing, META 2010*. D’Jerba (Tunissia), pp. 1–2

J. Toutouh and E. Alba (2011b). “Optimizing OLSR in VANETs with Differential Evolution: A Comprehensive Study”. In: *First ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet ’11)*. Florida, USA: ACM, pp. 1–8

J. Toutouh and E. Alba (2012b). “Multi-objective OLSR optimization for VANETs”. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*. Barcelona, Spain, pp. 571–578

J. Toutouh and E. Alba (2012c). “Parallel Swarm Intelligence for VANETs Optimization”. In: *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012 Seventh International Conference on*. IEEE, pp. 285–290

J. Toutouh and E. Alba (2013b). “Optimizing Telecommunications in Vehicular Networks with a Parallel Multiobjective PSO”. in: *22nd International Conference on Multiple Criteria Decision Making (MCDM2013)*, p. 295

R. Massobrio, J. Toutouh, and S. Nesmachnow (2015a). “A Multiobjective Evolutionary Algorithm for Infrastructure Location in Vehicular Networks”. In: *7th European Symposium on Computational Intelligence and Mathematics (ESCIM 2015)*. Cádiz, Spain, pp. 1–6

R. Massobrio, S. Bertinat, J. Toutouh, S. Nesmachnow, and E. Alba (2015b). “Smart placement of RSU for vehicular networks using multiobjective evolutionary algorithms”. In: *2nd LA-CCI Congress on Computational Intelligence (LA-CCI 2015)*. Curitiba, Brazil, pp. 1–6

National conferences [3]:

J. Toutouh, J. García-Nieto, and E. Alba (2010a). “Configuración Óptima del Protocolo de Encaminamiento OLSR para VANETs Mediante Evolución Diferencial”. In: *Actas del VII Congreso sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB’10)*. Valencia (Spain), pp. 463–471

J. Toutouh and E. Alba (2013a). “Computación Natural en Redes Vehiculares”. In: *Doctoral Consortium de Multiconferencia CAEPIA - XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA’13)*. Madrid (Spain), pp. 1740–1745

J. Toutouh and E. Alba (2015a). “Comunicación efficiente entre vehículos aplicando un algoritmo multi-objetivo paralelo”. In: *Actas del VII Congreso sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB’2015)*. Mérida (Spain), pp. 503–510

Appendix B

Complementary Results

B.1 Data Transfer Between Vehicles with Optimized QoS Results

This section includes different tables with complementary results obtained during the off-line optimization of the VDTP protocol (see Section 4.2).

B.1.1 Parameterization of the Used NC Algorithms

Table B.1 shows the results obtained in the preliminary parameters tuning procedure presented in Section 4.2.4.

TABLE B.1: Different combinations and results of the preliminary parameter tuning of the algorithms used in FTC optimization problem.

Algorithm	Parameter / Instances	Values / Results				
PSO	φ_1	2.0	2.0	2.0	2.0	2.0
	φ_2	2.0	2.0	2.0	2.0	2.0
	w	0.1	0.3	0.5	0.7	0.9
	Urban	1.952	1.978	**1.634**	2.766	3.280
	Highway	5.676	4.622	**4.1761**	5.283	6.045
DE	Cr	0.1	0.3	0.5	0.7	0.9
	μ	0.9	0.7	0.5	0.3	0.1
	Urban	4.027	2.647	2.241	1.866	**1.742**
	Highway	7.255	5.622	4.776	4.734	**4.663**
GA	P_{cros}	0.2	0.4	0.6	0.8	1.0
	P_{mut}	0.8	0.6	0.4	0.2	0.1
	Urban	2.701	2.245	1.953	**1.908**	2.077
	Highway	5.216	4.848	**4.380**	4.490	4.609
ES	P_{cros}	0.1	0.3	0.5	0.7	0.9
	P_{mut}	0.9	0.7	0.5	0.3	0.1
	Urban	4.920	3.878	3.031	2.606	**2.151**
	Highway	7.836	6.877	6.240	**5.783**	5.923
SA	T	0.2	0.4	0.6	0.8	1.0
	Urban	4.922	1.978	2.785	**1.634**	3.744
	Highway	7.665	5.201	4.820	**4.424**	4.683

B.1.2 Scalability Analysis

Table B.2 presents the results of the scalability analysis experimentation described in Section 4.2.4. The three last columns of the table show the time required to find the best solution (T_{best}) for each VANET instance.

TABLE B.2: Performance comparison in terms of average fitness and average optimization time (T_{best}) of the scaled *Urban* VANET scenarios.

Algorithm	Average fitness ± Stdev.			T_{best}		
	Scenario U1	Scenario U2	Scenario U3	Scenario U1	Scenario U2	Scenario U3
PSO	**1.6346±17.74%**	**1.3920±20.34%**	**3.6763±12.06%**	7.95E+03	5.93E+03	1.20E+04
DE	1.7423±21.33%	1.4504±13.00%	3.9186±18.93%	7.12E+03	1.10E+04	1.43E+04
GA	1.9086±11.84%	1.4100±8.76%	3.6829±13.75%	6.68E+03	9.81E+03	1.41E+04
ES	2.1517±5.88%	1.5462±38.95%	3.7799±16.47%	9.00E+03	8.99E+03	1.50E+04
SA	2.7850±31.30%	2.3880±42.74%	3.8143±3.30%	**4.76E+03**	**3.40E+03**	**5.36E+03**

B.1.3 Final Fitness Values Statistical Test results

Table B.3 shows the Friedman ranking of the compared algorithms in *Urban* and *Highway* instances (the best ranked algorithm is in the top).

TABLE B.3: Friedman Rank test results of NC solving FTC (confidence level set to 99%).

Urban		Highway	
Algorithm	Rank	Algorithm	Rank
PSO	**1.27**	SA	**1.87**
DE	1.83	GA	1.97
GA	3.07	PSO	2.63
ES	4.33	DE	3.57
SA	4.50	ES	4.97

Tables B.4 and B.5 contain the resulted *p-value* of applying the Wilcoxon Signed Rank test with a confidence level of 99% to PSO (the best ranked one for *Urban* instance) and SA (the best ranked one for *Highway* instance) in comparison with the remaining of algorithms, respectively. In this tables, the symbol ▲ means that there are statistical differences between the algorithms, and therefore, the best ranked algorithm is significantly better than the compared algorithm, whereas the symbol △ means that non-statistical difference can be assured between the two algorithms.

TABLE B.4: PSO versus others Wilcoxon Signed Rank test in FTC (*Urban* scenario).

Algorithm	Test	*p-value*
DE	▲	0.001
GA	▲	<0.001
ES	▲	<0.001
SA	▲	<0.001

TABLE B.5: SA versus others Wilcoxon Signed Rank test in FTC (*Highway* scenario).

Algorithm	Test	*p-value*
PSO	△	0.371
DE	▲	<0.001
GA	△	0.975
ES	▲	<0.001

B.2 Optimization of the QoS of Proactive Routing Results

This section includes different tables with complementary results obtained during the QoS off-line optimization of the OLSR routing protocol (see Section 4.3).

B.2.1 Parameterizations Analyzed of QoS Optimization of OLSR

Table B.6 the OLSR parameter settings considered for comparison in the analysis in Section 4.3.5. Columns 2 to 4 contain three human expert configurations (#1, #2, and #3) proposed by Gómez et al. (2005); columns 5 and 6 contain the OLSR configurations of the RFC 3626 and the one obtained by the random search, respectively; columns 7 to 10 show the best OLSR configurations obtained by each one of the NC algorithms analyzed: PSO, DE, GA, and SA.

TABLE B.6: OLSR parameterizations of the state of the art (Gómez et al. (2005)), the standard RFC 3626, and the best solutions in optimization algorithms validated in Section 4.3.5.

Metric	Human experts			OLSR	Optimized configurations				
	#1	#2	#3	RFC	RAND	DE	PSO	GA	SA
HELLO_INTERVAL	0.50	1.0	4.0	2.0	3.730	8.477	8.909	8.568	9.005
REFRESH_INTERVAL	0.50	1.0	4.0	2.0	6.188	1.086	9.663	15.829	4.925
TC_INTERVAL	1.25	2.5	10.0	5.0	5.188	7.246	7.192	5.286	6.753
WILLINGNESS	3	3	3	3	4	0	1	1	0
NEIGHB_HOLD_TIME	1.50	3.0	12.0	6.0	5.400	16.924	67.238	83.771	80.334
TOP_HOLD_TIME	3.75	7.5	20.0	15.0	40.164	99.061	72.693	67.619	80.965
MID_HOLD_TIME	3.75	7.5	20.0	15.0	34.476	6.713	91.303	37.105	2.913
DUP_HOLD_TIME	30.0	30.0	30.0	30.0	31.515	71.938	21.572	16.268	16.705

B.2.2 Results of the Validation Experiments of QoS Optimization of OLSR

Table B.7 presents for each OLSR configuration found using the optimization algorithms, the median values for each studied metric, computed in the simulations performed over the 54 different VANET scenarios. The results are compared with the values obtained in simulations performed with the standard OLSR configuration suggested by RFC 3626. The best median values obtained for each metric are marked in bold.

TABLE B.7: Median results of the validation of the QoS optimized OLSR configurations.

Scenario	Configurations	PDR	NRL	E2ED	RPL
U1	SA	**99.95%**	15.09%	2.13 ms	1.03
	DE	92.58%	**12.64%**	4.34 ms	1.09
	GA	**99.95%**	16.95%	**2.10 ms**	**1.01**
	PSO	99.39%	12.73%	2.60 ms	1.05
	RAND	94.41%	18.35%	17.16 ms	1.38
	RFC	99.40%	22.28%	2.79 ms	1.05
U2	SA	84.01%	**12.36%**	8.99 ms	1.59
	DE	85.77%	10.04%	9.23 ms	1.54
	GA	85.70%	15.82%	**3.81 ms**	1.63
	PSO	**86.03%**	**12.36%**	10.56 ms	1.54
	RAND	84.98%	20.15%	21.11 ms	1.41
	RFC	85.91%	23.94%	8.27 ms	**1.18**
U3	SA	74.85%	10.05%	44.38 ms	1.29
	DE	78.29%	**9.07%**	**19.19 ms**	1.31
	GA	75.08%	11.17%	44.95 ms	1.30
	PSO	75.05%	9.97%	45.81 ms	1.28
	RAND	69.76%	15.58%	398.42 ms	1.49
	RFC	**86.71%**	20.65%	70.26 ms	**1.05**
Global Results	SA	84.76%	14.56%	**4.04 ms**	1.35
	DE	84.29%	**11.98%**	10.24 ms	1.34
	GA	87.85%	16.32%	4.36 ms	1.34
	PSO	86.73%	12.73%	8.12 ms	1.46
	RAND	88.93%	19.21%	17.16 ms	1.38
	RFC	**89.56%**	23.15%	6.06 ms	**1.09**

B.3 Power-aware of Proactive Routing Results

This section includes different tables with complementary results obtained during the off-line optimization of the energy-efficiency OLSR routing protocol (see Section 4.4).

B.3.1 Parameterization of the Energy-efficiency Optimization of OLSR

Table B.8 presents the results for the combinations of p_C and p_M analyzed, reporting the average, relative standard deviation, and best values of fitness; the average energy and PDR, and the average gaps in energy and PDR with the standard RFC configuration.

TABLE B.8: Results of parameter setting of the pGA for solving power-aware optimization of OLSR.

(p_C, p_M)	Fitness			Metrics		GAP RFC	
	Avg.	Stdev.	Best	Energy	PDR	Energy	PDR
(0.5,0.06125)	0.576836	0.31%	0.572319	3454.40	75.03%	39.19%	-14.95%
(0.7,0.06125)	0.577790	0.55%	0.571034	3446.11	75.01%	39.34%	-14.99%
(0.9,0.06125)	0.577498	0.39%	0.572754	3459.03	75.20%	39.11%	-14.77%
(0.5,0.125)	0.573733	0.21%	0.571268	3447.76	75.03%	39.31%	-14.95%
(0.7,0.125)	0.573778	0.24%	0.570946	3445.84	75.05%	39.34%	-14.93%
(0.9,0.125)	0.576217	0.14%	0.574546	3470.34	75.33%	38.91%	-14.61%
(0.5,0.25)	0.574279	0.13%	0.572724	3457.23	75.01%	39.14%	-14.99%
(0.7,0.25)	0.572346	0.15%	0.570118	3440.33	75.01%	39.44%	-14.99%
(0.9,0.25)	0.572408	0.17%	0.570351	3442.20	75.07%	39.41%	-14.91%

B.3.2 Statistical Analysis of pGA

Table B.9 shows the results of the Kurskal-Wallis statistical test applied over the fitness results obtained by the pGAs solving power-aware optimization of OLSR.

TABLE B.9: Statistical analysis of pGAs results in addressing power-aware optimization of OLSR.

Statistical test		Algorithm		
		pGA-8	pGA-16	pGA-24
Kruskal-Wallis	pGA-8	-	6.4×10^{-4}	1.9×10^{-7}
	pGA-16	6.4×10^{-4}	-	0.015
	pGA-24	1.9×10^{-7}	0.015	-

B.3.3 Results of Validation Experiments of Power-Aware OLSR

The validation analysis evaluated several metrics related to the energy-aware and QoS of the communication. From the point of view of the power consumption, the energy in transmitting (E_{send}) and receiving (E_{recv}) mode, as well as the total energy (E_{total}) and total energy per vehicle ($E_{tot\times v}$) were studied. From the point of view of QoS, the studied metrics include the PDR, the E2ED (in miliseconds), the NRL, and the RPL . Table B.10 presents for each best OLSR configuration found using the three pGAs studied, the average values for each studied metric, computed in the simulations performed over the 36 VANET scenarios.

Table B.11 summarizes the results of the Friedman and Wilcoxon statistical tests regarding the energy gaps. In the Wilcoxon test, the group of three values reported corresponds to the positive ranks, average positive ranks, and the sum of positive ranks for every pairwise comparison, respectively.

TABLE B.10: Results of the validation experiments of energy-efficient OLSR configurations.

Configuration	Energy metrics				QoS metrics			
	E_{sent}	E_{recv}	E_{total}	$E_{tot \times v}$	PDR	E2ED	NRL	RPL
				Medium size (U2)				
pGA-8	12099.05	5265.45	17364.49	604.12	61.54%	62.39	3.36%	1.58
pGA-16	11902.02	5206.53	17108.55	589.17	63.64%	58.35	3.53%	1.43
pGA-24	11776.50	5094.87	16871.36	575.86	61.80%	55.04	3.34%	1.47
RFC	17918.45	8102.75	26021.20	876.91	70.22%	1356.18	25.46%	1.25
				Large size (U3)				
pGA-8	14682.85	7030.52	21713.36	491.22	55.75%	505.30	3.98%	1.50
pGA-16	14864.78	7120.72	21985.51	505.51	57.63%	490.34	3.73%	1.48
pGA-24	14249.18	6762.22	21011.39	479.16	56.65%	483.62	3.57%	1.45
RFC	21574.81	16247.10	37821.93	877.75	64.00%	868.57	28.34%	1.15
				Overall				
pGA- 8	13390.95	6147.99	19538.93	547.67	58.64%	283.85	3.67%	1.54
pGA-16	13383.40	6163.63	19547.03	547.34	60.64%	274.34	3.63%	1.46
pGA-24	**13012.84**	**5928.54**	**18941.37**	**527.51**	59.22%	**269.33**	**3.45%**	1.46
RFC	19572.25	12102.03	31674.29	877.33	**67.89%**	506.26	25.22%	**1.20**

TABLE B.11: Statistical analysis of the energy results.

Statistical test		**Configuration**			
		pGA-8	pGA-16	pGA-24	RFC
Friedman (*Avg. rank*)		2.19	1.94	1.92	3.94
Wilcoxon	pGA-8	-	(14, 19.8, 277)	(16, 16.6, 266)	(35, 19.0, 665)
	pGA-16	(22, 17.7, 389)	-	(16, 17.1, 274)	(36, 18.5, 666)
	pGA-24	(20, 20.0, 400)	(20, 19.6, 392)	-	(35, 18.9, 661)
	RFC	(19.0, 1.0, 1)	(18.5, 0.0, 0)	(1, 5.0, 5)	-

B.4 Multi-objective Optimization of QoS Routing Results

This section includes different tables with complementary results obtained during the multi-objective optimization of the QoS of AODV routing protocol (see Section 4.5).

B.4.1 Parameter Configuration of pMOAs for addressing AODV MO-QoS

Table B.12 presents the median hypervolume value for each parameterization analyzed for pNSGA-II and pSMPSO.

TABLE B.12: Median hypervolume value for each parameterization of pNSGA-II and pSMPSO.

p_C		p_M			
		$\frac{1}{4L}$=0.023	$\frac{1}{2L}$=0.045	$\frac{1}{L}$=0.091	$\frac{1}{0.5L}$=0.182
pNSGA-II	0.3	0.757	0.749	0.751	0.801
	0.5	0.734	0.760	0.810	0.776
	0.7	0.752	0.776	0.786	0.810
	0.9	**0.832**	0.770	0.767	0.776
pSMPSO		0.738	0.755	**0.758**	0.747

B.4.2 Optimized AODV Configurations

Table B.13 shows the optimized AODV configurations evaluated in Section 4.5.5.

TABLE B.13: The AODV parameterizations obtained by the pMOAs evaluated in this study.

AODV parameter	pNSGA-II	pSMPSO
`HELLO_INTERVAL`	10.46	3.94
`ACTIVE_ROUTE_TIMEOUT`	10.55	2.14
`MY_ROUTE_TIMEOUT`	20.42	8.06
`NODE_TRAVERSAL_TIME`	6.89	10.00
`MAX_RREQ_TIMEOUT`	41.13	40.62
`NET_DIAMETER`	21	24
`ALLOWED_HELLO_LOSS`	6	1
`REQ_RETRIES`	6	1
`TTL_START`	7	19
`TTL_INCREMENT`	3	8
`TTL_THRESHOLD`	19	5

B.4.3 Validation Experiments of the QoS Optimized AODV Configurations

Table B.14 shows the results of the validation experiments presented in Section 4.5.5.

TABLE B.14: Median values of each QoS metric and each analyzed AODV configuration grouped by scenario area size.

Configurations	QoS metrics		
	PDR	**NRL**	**E2ED**
Small size VANET scenario (U1)			
RFC	64.127	82.101	69.896
pPSO	69.745	57.331	42.894
GN	67.811	61.079	**56.244**
pNSGA-II	66.270	79.659	102.798
pSMPSO	**71.788**	**51.336**	60.085
Medium size VANET scenario (U2)			
RFC	**83.465**	10.829	20.183
pPSO	71.188	**7.689**	15.131
GN	72.542	8.434	**15.028**
pNSGA-II	81.758	8.466	36.512
pSMPSO	78.125	8.810	31.389
Large size VANET scenario (U3)			
RFC	27.979	18.337	120.692
pPSO	25.761	**11.517**	15.875
GN	23.622	12.209	**5.861**
pNSGA-II	25.153	18.434	249.932
pSMPSO	**29.362**	11.777	118.328
Overall			
RFC	64.510	59.269	59.787
pPSO	59.269	**12.320**	17.183
GN	59.787	13.063	**10.753**
pNSGA-II	67.015	19.416	97.206
pSMPSO	**69.203**	12.401	65.681

B.5 On-line Broadcasting Optimization Results

Table B.15 shows the Friedman ranking of the Swarm FREEDY algorithms (see Section 5.6) in terms of channel occupancy and network balance (the best ranked algorithms are in the top). Please notice that, in case of occupancy, the higher values represent the better results, but in case of balance, the lower values represent the better results.

TABLE B.15: Friedman Rank test results of the Swarm FREEDY algorithms devised in our study (confidence level set to 99%).

Occupancy		Balance	
Algorithm	Rank	Algorithm	Rank
Swarm *o*-FREEDY-med	**2.99**	**Swarm *n*-FREEDY-mod**	**1.42**
Swarm *n*-FREEDY-med	2.67	Swarm *n*-FREEDY-med	2.31
Swarm *n*-FREEDY-mod	2.64	Swarm *o*-FREEDY-mod	2.82
Swarm *o*-FREEDY-mod	1.70	Swarm *o*-FREEDY-med	3.45
p-value $= 4.01 \times 10^{-32}$		*p-value* $= 3.66 \times 10^{-78}$	

Tables B.16 and B.17 present the Wilcoxon test results in terms of *p-value* of comparing all the Swarm FREEDY algorithms between each other in order to confirm the Friedman ranking results.

TABLE B.16: Post hoc statistical results of Swarm FREEDY algorithms (occupancy). The table shows the *p-values* of applying Wilcoxon test.

Statistical test		Algorithms			
		Sw *o*-FREEDY-med	Sw *o*-FREEDY-mod	Sw *n*-FREEDY-med	Sw *n*-FREEDY-mod
Wilcoxon	Swarm *o*-FREEDY-med		4.57×10^{-27}	3.13×10^{-3}	**0.99**
	Swarm *o*-FREEDY-mod	4.57×10^{-27}		2.42×10^{-8}	6.07×10^{-27}
	Swarm *n*-FREEDY-med	3.13×10^{-3}	2.42×10^{-8}		3.01×10^{-6}
	Swarm *n*-FREEDY-mod	**0.99**	6.07×10^{-27}	3.01×10^{-6}	

Regarding the occupancy of the channel, the first and second ranked algorithms (i.e., Swarm *o*-FREEDY-med and Swarm *n*-FREEDY-med, respectively) are not statistically different since the computed *p-value* when comparing them is 0.99 (>0.01). Therefore, Swarm *o*-FREEDY-med cannot be considered better than Swarm *n*-FREEDY-med. The other algorithms are significantly different between each other.

TABLE B.17: Post hoc statistical results of Swarm FREEDY algorithms (balance). The table shows the *p-values* of applying Wilcoxon test.

Statistical test		Algorithms			
		Sw *o*-FREEDY-med	Sw *o*-FREEDY-mod	Sw *n*-FREEDY-med	Sw *n*-FREEDY-mod
Wilcoxon	Swarm *o*-FREEDY-med		**0.43**	5.09×10^{-46}	4.92×10^{-46}
	Swarm *o*-FREEDY-mod	**0.43**		6.20×10^{-8}	7.73×10^{-24}
	Swarm *n*-FREEDY-med	5.09×10^{-46}	6.20×10^{-8}		4.28×10^{-33}
	Swarm *n*-FREEDY-mod	4.92×10^{-46}	7.73×10^{-24}	4.28×10^{-33}	

Regarding the balance of the network, the first ranked algorithm (i.e., Swarm *n*-FREEDY-med) is significantly the best since the computed *p-values* are lower than 0.01 when comparing it with all the other analyzed algorithms.

B.6 Real World Test Results

Table B.18 shows the results of the real outdoor testbed when using personal portable devices in terms of the average E2ED in milliseconds (ms) of each data type and for each one of communication schemes analyzed in Section 7.3.4. The results are grouped by the distance between the cars during the data transfers. The cases in which none of the data packets sent reached the destination node are shown in the table by a dash (-).

TABLE B.18: Experimental results of the outdoor test by using portable personal devices in terms of the average V2V E2ED (milliseconds) grouped by distance between vehicles (separation) and the data packet type (size in bytes).

Connection type	Data size in bytes					
	32	64	128	256	512	Avg.
	separation = 25 meters					
sph-sph-25m	77.1	108.9	173.8	171.2	87.0	116.8
sph-tab-25m	25.8	22.2	25.5	29.5	23.7	25.2
sph-laptG-25m	18.6	8.8	7.7	8.8	9.1	10.0
tab-tab-25m	30.0	20.9	71.6	47.7	51.7	40.6
tab-laptG-25m	17.7	6.2	7.4	10.6	7.1	9.1
laptG-laptG-25m	1.2	1.4	0.9	2.1	1.8	1.4
laptA-laptA-25m	8.7	9.2	12.0	14.8	16.7	12.3
	separation = 50 meters					
sph-sph-50m	83.8	107.0	159.0	83.0	90.8	101.4
sph-tab-50m	23.2	29.0	26.6	54.4	23.3	29.6
sph-laptG-50m	29.2	9.0	9.2	21.2	21.9	16.2
tab-tab-50m	27.7	51.5	40.2	38.8	126.0	48.9
tab-laptG-50m	58.8	20.8	43.5	45.1	73.9	44.6
laptG-laptG-50m	1.7	0.8	1.6	2.1	1.8	1.6
laptA-laptA-50m	13.5	14.6	15.8	29.5	16.9	18.1
	separation = 75 meters					
sph-sph-75m	23.4	488.0	1042.0	158.0	187.0	203.8
sph-tab-75m	32.3	26.4	36.1	41.3	51.7	36.6
sph-laptG-75m	15.1	10.4	11.0	9.8	22.0	13.0
emphtab-tab-75m	90.5	55.1	98.2	85.5	76.9	79.7
tab-laptG-75m	33.2	8.9	7.8	10.5	8.7	11.6
laptG-laptG-75m	2.2	1.6	2.5	2.1	2.2	2.1
laptA-laptA-75m	14.6	19.6	14.6	15.5	17.7	16.5
	separation = 100 meters					
sph-sph-100m	-	-	-	-	-	-
sph-tab-100m	-	-	-	-	-	-
sph-laptG-100m	26.0	11.4	13.0	16.1	13.3	15.2
tab-tab-100m	56.1	220.1	165.3	561.1	134.2	172.7
tab-laptG-100m	66.7	55.2	78.5	149.7	26.3	64.7
laptG-laptG-100m	2.8	3.2	7.0	5.7	6.9	4.8
laptA-laptA-100m	96.2	146.0	154.9	207.1	168.7	154.6
	separation = 125 meters					
sph-sph-125m	-	-	-	-	-	-
sph-tab-125m	-	-	-	-	-	-
sph-laptG-125m	26.4	8.6	9.5	17.7	14.2	14.0
tab-tab-125m	51.3	83.6	216.4	122.5	182.1	115.7
tab-laptG-125m	27.2	23.3	10.2	7.5	13.5	14.6
laptG-laptG-125m	1.6	1.7	1.5	2.0	2.7	1.9
laptA-laptA-125m	-	-	-	-	-	-
	separation = 150 meters					
sph-sph-150m	-	-	-	-	-	-
sph-tab-150m	-	-	-	-	-	-
sph-laptG-150m	8.6	10.3	21.0	15.6	11.2	12.7
tab-tab-150m	52.4	290.7	-	-	-	-
tab-laptG-150m	132.0	175.1	78.2	73.2	151.0	114.8
laptG-laptG-150m	2.0	1.8	1.7	3.6	4.9	2.6
laptA-laptA-150m	-	-	-	-	-	-

Appendix C

Resumen en Español

La aplicación de los últimos avances en tecnologías de la información y las comunicación (TIC) a entornos vehiculares ha resultado en la aparición de las **redes vehiculares *ad hoc* o VANETs** (*vehicular ad hoc networks*). Estas redes de comunicación inalámbrica sin infraestructura se forman de forma espontánea, principalmente entre vehículos cercanos y distintos elementos de la infraestructura vial como semáforos o sensores. Así, estas redes ofrecen la posibilidad de desarrollar aplicaciones revolucionarias en el ámbito de la seguridad y la eficiencia vial mediante el intercambio continuo de información relevante sobre tráfico. Sin embargo, el dinamismo y las limitaciones de las tecnologías inalámbricas utilizadas (basadas en IEEE 802.11) abren un nuevo conjunto de problemas de gran complejidad que tienen que ser investigados previamente si se quiere una implantación eficiente de dicha tecnología.

Ya existe un importante campo de conocimiento en el ámbito de las comunicaciones móviles *ad hoc* o MANETs (*mobile ad hoc networks*), y en un principio se han tratado de implantar estas soluciones en las VANETs. Sin embargo, estas propuestas no han ofrecido el rendimiento necesario. Así, ha surgido un cuerpo del conocimiento específico, que debido a la importancia que pueden tener las VANETs en la sociedad actual, está creciendo y evolucionando continuamente. Al ser un dominio tan novedoso, existe una serie de cuestiones abiertas, como el encaminamiento y la difusión de paquetes de datos, que todavía no han podido resolverse empleando estrategias clásicas. Es por tanto necesario crear y estudiar nuevas técnicas que permitan de forma eficiente, eficaz, robusta y flexible resolver dichos problemas.

A lo largo de la historia, la naturaleza ha sido fuente de inspiración para diseñar desde las herramientas más rudimentarias a procesos computacionales complejos. Este trabajo de tesis doctoral propone el uso de computación inspirada en la naturaleza o **Computación Natural (CN)** para tratar algunos de los problemas más importantes en el ámbito de las VANETs. Además de resolver los problemas VANET en los que nos enfocamos, se han realizado avances en el uso de estas técnicas para que traten estos problemas de forma más eficiente y eficaz. Por último, se han llevado a cabo pruebas reales de concepto empleando vehículos y dispositivos de comunicación reales en la ciudad de Málaga (España).

C.1 Organización

Esta tesis doctoral se divide en tres grandes bloques. El primer bloque presenta los principales fundamentos en los que se basa la tesis: primero se introduce el dominio de las redes vehiculares y después se presenta la Computación Natural como herramienta eficiente para resolver problemas de optimización complejos, como los que se han tratado en este dominio de las VANETs. El segundo bloque, el de mayor extensión, detalla la resolución de los problemas de optimización en VANETs en los que se ha profundizado en la tesis y las pruebas reales de concepto. Los problemas tratados son los siguientes: la diseminación eficiente de datos, teniendo en cuenta tres aspectos distintos (la *transferencia de archivos*, el *encaminamiento o routing* de paquetes y la *difusión o broadcasting* de mensajes) y el *diseño inteligente de la infraestructura de estaciones base o RSUs (RSU-DP)*. Finalmente, el último bloque agrupa los principales logros alcanzados en esta tesis y se extraen las conclusiones. A continuación detallamos de manera específica el contenido por capítulos.

- **Capítulo 1: Introducción.** En este capítulo se realiza una justificación de las razones que han motivado el trabajo que incluye la presente tesis, y se esboza un esquema del contenido de la misma.

- **Capítulo 2: Redes Vehiculares.** Este capítulo describe de manera general las VANETs. Presentamos aquí distintos aspectos de las tecnologías de comunicación que se aplican, así como la amplia aplicabilidad de estas redes y las principales diferencias con otros tipos de redes móviles *ad hoc*. Finalmente, se revisa el panorama general sobre los principales proyectos y consorcios que tratan o han tratado el tema de las VANETs a nivel mundial, así como se revisa las principales cuestiones abiertas en las que actualmente se están trabajando.

- **Capítulo 3: Computación Natural y Optimización en VANETs.** Esta capítulo introduce el campo de la CN poniendo especial atención en su aplicación en el ámbito de la resolución de problemas de optimización complejos. A continuación, se especifican los problemas de optimización que se tratan en la tesis, así como se revisa la literatura relativa a su resolución. Finalmente, se incluye la metodología aplicada para evaluar tanto las soluciones obtenidas como las técnicas empleadas para obtenerlas.

- **Capítulo 4: Optimización *off-line* de protocolos.** Este capítulo describe el problema de optimización *off-line* de protocolos para VANET, es decir, la búsqueda de la configuración de los parámetros de los protocolos que optimice su rendimiento. Se han analizado tanto protocolos de transferencia de archivos como de encaminamiento de paquetes. Se han estudiado diferentes tipos de formulación mono-objetivo y multi-objetivo. El análisis experimental incluye el uso de diferentes técnicas CN que se han mejorado mediante la aplicación de operadores específicos y paralelizando los algoritmos para que la búsqueda requiera menos coste computacional. Finalmente, se estudia la eficacia de los resultados obtenidos mediante simulaciones VANET realistas basadas en información de Málaga.

- **Capítulo 5: Optimización *on-line* del *broadcasting*.** Aquí se presenta el problema de congestión del medio debido a la difusión de mensajes (*beacons*) en VANETs. Después se proponen cuatro tipos de algoritmos voraces (*greedy*) dinámicos para la difusión eficiente de mensajes (familia de algoritmos *FREEDY*). Finalmente, hemos realizado unos experimentos de validación comparando estos algoritmos con otros del estado del arte.

- **Capítulo 6: Diseño inteligente de la infraestructura.** Este capítulo trata el problema del diseño de infraestructura (RSU-DP) de forma novedosa, empleando una formulación multi-objetivo. Así, se optimizan de forma conjunta los dos objetivos contrapuestos de este problema (maximizar el servicio que se ofrece y minimizar el coste del despliegue). El problema se ha resuelto sobre una instancia realista de Málaga empleando un algoritmo evolutivo paralelo multi-objetivo, que utiliza operadores originales.

- **Capítulo 7: Pruebas reales.** En este capítulo se presentan las pruebas reales de concepto en las que se han utilizado vehículos en movimiento por las carreteras de Málaga. Se han realizado dos tipos de pruebas distintas: en las primeras, se ha comprobado cómo un protocolo mejorado mediante CN ofrece un mejor rendimiento que su versión estándar; en las segundas, se ha estudiado el uso de dispositivos móviles personales (teléfonos inteligentes, tabletas y ordenadores portátiles) para comunicaciones VANET.

- **Capítulo 8: Conclusiones y trabajo futuro.** Aquí se resumen las principales conclusiones extraídas de trabajo realizado y se presentan distintas directrices para el trabajo futuro.
- **Apéndice A: Publicaciones.** En este apéndice se listan las publicaciones realizadas como consecuencia del trabajo enmarcado dentro de la presente tesis doctoral.
- **Apéndice B: Tablas y resultados detallados.** En este apéndice se incluyen tablas y resultados de distintos experimentos realizados en el marco de la tesis, que no han podido incluirse en el manuscrito principal.
- **Apéndice C: Resumen en español.** El presente resumen de la tesis.

C.2 Redes Vehiculares

El continuo crecimiento de la población mundial y su concentración cada día más acusada en ciudades ponen en jaque la viabilidad del modelo de desarrollo urbano. La iniciativa mundial *Smart Cities* persigue mitigar este problema incrementando la calidad de vida de los ciudadanos, mejorando la eficiencia de los recursos, facilitando la participación ciudadana, y, en definitiva, garantizando el desarrollo sostenible de las mismas. Uno de los ejes fundamentales es la *Smart Mobility*, que trata de paliar los problemas ocasionados por la congestión de las carreteras permitiendo desplazamientos más seguros, cómodos y eficientes.

Así, han aparecido diversas iniciativas impulsadas por los gobiernos y por la industria, desarrollándose nuevas soluciones en el campo de los Sistemas Inteligentes de Transporte (SIT). Fruto de estas iniciativas ha sido el diseño de servicios orientados a la prevención de accidentes, a la mejora de la eficiencia (tiempos de desplazamiento, emisiones de CO_2, etc.) e incluso al entretenimiento de los pasajeros.

Uno de los pilares de estos nuevos sistemas son las redes vehiculares *ad hoc* o VANETs (*vehicular ad hoc networks*). Las VANETs son redes descentralizadas que proveen de una plataforma para el diálogo de vehículos entre sí (*vehicle-to-vehicle* -V2V-) y con los elementos de la infraestructura vial (*vehicle-to-infrastructure* -V2I-) como semáforos, señales de tráfico, etc. Así, los vehículos pueden recibir, procesar y difundir información actualizada sobre distintos aspectos del tráfico (ver Figura C.1). En las últimas propuestas también se han incluido comunicaciones vehiculares empleando redes de amplia cobertura como WiMAX o redes celulares (*vehicle-to-broadband* -V2B-), aunque solo para cierto tipo de servicios (Hartenstein y Laberteaux, 2009; Campolo y col., 2015).

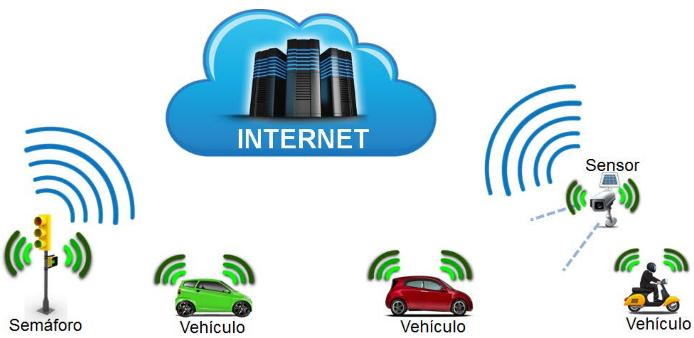


FIGURA C.1: Escenario VANET típico.

El estándar IEEE 802.11p (ETSI, 2010), basado en comunicaciones directas de corto alcance (DSRC), se ha definido expresamente para el acceso al medio inalámbrico en entornos vehiculares (WAVE) (Uzcategui y Acosta-Marum, 2009). A pesar de su existencia, en la actualidad se están empleando otros estándares más generalizados para las primeras pruebas de concepto, como puede ser el IEEE 802.11g. La limitada cobertura de los estándares de acceso al medio que se está utilizando (basados en IEEE 802.11), la existencia de interferencias/obstáculos y la alta movilidad de los nodos provoca que los enlaces que se crean durante la comunicación tengan un tiempo de vida muy limitado, lo que complica de forma crítica el correcto intercambio de paquetes (frecuentes cambios de topología y fragmentación de la red). Así, el **encaminamiento** (*routing*) eficiente de paquetes en redes vehiculares (totalmente descentralizadas) es una tarea altamente compleja (Lee y col., 2010; Chen y col., 2011). Este problema también afecta a los protocolos de comunicaciones de capas más altas, como los protocolos de transporte de datagramas o **transferencia de archivos**.

Además, mucha de las aplicaciones para VANETs requieren de un servicio continuo de **difusión** (*broadcasting*) de mensajes para el descubrimiento de nodos cercanos y el envío de información cinemática de los vehículos (velocidad, aceleración, posición, etc.) (Sengupta y col., 2007). Sin embargo, en situaciones de tráfico denso (número elevado de conexiones), aparecen serios problemas de congestión y colisión debido principalmente a la retransmisión repetida de mensajes sobre un mismo canal (problema de tormenta por difusión) (Chen y col., 2010; Sattari y col., 2012).

Cabe destacar la importancia en este campo de disponer de una plataforma física para el despliegue de este tipo de redes, es decir, de una **infraestructura** compuesta por nodos fijos, conocidos como estaciones base o RSU (*roadside units*). Estos se emplean para comunicar nodos móviles con los elementos de la infraestructura vial, con redes estáticas (Internet) con otros nodos móviles que estén fuera del alcance directo. La dificultad del diseño e implantación de este tipo de infraestructura consiste en la selección óptima de la localización de las estaciones base (podrían ser semáforos o farolas), así como de sus componentes hardware y software (Trullols y col., 2010). Con ello se persigue reducir los costes de implantación (económicos, sociales y medioambientales), maximizando la calidad del servicio (minimizar los problemas por congestión, maximizar la cobertura, etc.) y la robustez (alta disponibilidad).

Con los avances y la expansión de las diferentes redes de comunicación inalámbrica, como WiMAX, las redes celulares (3G, LTE o 4G) y otros, surge la posibilidad de ofrecer comunicación ubicua a los vehículos (Hossain y col., 2010). Sin embargo, la **selección eficiente de qué tecnología de acceso** al medio utilizar para enviar un determinado mensaje no es sencilla, puesto que depende de muchos factores como la aplicación que quiere comunicarse (requisitos) o el estado de la red que se quiere utilizar (número de nodos conectados, etc.).

Esta tesis doctoral trata de resolver estos cuatro problemas en el ámbito de las VANETs. Para ello los trata como problemas de optimización (complejos) que se van a resolver empleando modernas técnicas de optimización basadas en CN.

C.3 Computación Natural y Optimización

La naturaleza ha inspirado al ser humano desde el inicio de los tiempos en el diseño de herramientas, procesos y objetos complejos en general. Así, la CN aparece como el desarrollo de sistemas computacionales extrayendo ideas de la naturaleza o utilizando medios naturales (De Castro, 2006). En esta tesis nos hemos centrado en la rama de la CN que se dedica al diseño de algoritmos para la resolución de problemas complejos (algoritmos bio-inspirados). Más concretamente, se han utilizado técnicas de CN que se engloban dentro de las *metaheurísticas* (Glover y Kochenberger, 2003; Blum y Roli, 2003) y que se agrupan en *Algoritmos Evolutivos* (EA), *Inteligencia de Enjambre* y *Enfriamiento Simulado* (SA).

Las metaheurísticas son estrategias de alto nivel (generalmente estocásticas) que combinan distintos métodos para explorar un espacio de búsqueda (normalmente extensos) de forma eficiente. Suelen definirse como métodos genéricos que se deben instanciar empleando información específica sobre problema al que se aplican (representación de las soluciones, operadores, etc.). Existen dos categorías de metaheurísticas según el número de soluciones sobre las que operan de forma simultánea: las basadas en trayectoria, que manejan una sola solución que mejoran de forma iterativa, como el SA; y las basadas en población, que operan sobre un conjunto de soluciones o población, como los EAs.

Dos son las principales razones que han motivado el uso de CN en la resolución de problemas en VANETs en esta tesis. La primera, y la principal, es que todos implican una gran complejidad computacional, por lo que requieren de la gestión eficiente de muchos recursos para su resolución. Esto se sufre principalmente en la diseminación eficiente de mensajes en VANETs (encaminamiento, difusión y transferencia de archivos). La segunda, y que ha aparecido en varios de los problemas, es que existen distintos objetivos en conflicto entre sí, es decir, que la mejora de uno de los objetivos implica el empeoramiento en los otros (Deb, 2001). Un ejemplo de esto se da en el diseño eficiente de la infraestructura, donde no se puede maximizar la calidad del servicio ofrecido por las RSUs y minimizar el coste de instalación al mismo tiempo. Este tipo de problemas requieren del uso de técnicas CN multi-objetivo que devuelven un conjunto de soluciones optimizadas no-dominadas (o frente de Pareto).

Cabe destacar que la generalización de plataformas de computación paralela y distribuida, ha llevado a los investigadores a utilizar dichas plataformas para mejorar la eficiencia de los métodos CN (en términos de tiempo de cómputo y memoria). Así, surgen las metaheurísticas paralelas (Alba, 2005). En este trabajo de tesis se ha empleado el modelo maestro-esclavo. Este modelo consiste en una división funcional del algoritmo, por lo que unas tareas las realiza el proceso maestro y otras se distribuyen por diversos elementos de proceso para que se ejecuten en paralelo. Específicamente, en los algoritmos propuestos en esta tesis se realiza en paralelo la evaluación de la calidad de la solución, que es la operación computacionalmente más costosa.

Para resolver los distintos problemas abordados en esta tesis se emplea un juego de algoritmos con distintas características. Estos son los siguientes:

- Algoritmos Evolutivos:
  - Mono-objetivo: Algoritmo Genético (GA), Estrategias Evolutivas (ES) y Evolución Diferencial (DE).
  - Multi-objetivo: GA basado en un Ordenamiento No-Elitista (NSGA-II)
- Inteligencia de Enjambre:
  - Mono-objetivo: Optimización por Cúmulo de Partículas (PSO)
  - Multi-objetivo: PSO Multi-objetivo con Velocidad Limitada (SMPSO)
- Enfriamiento Simulado (SA).

C.4 Problemas VANET Analizados

La hipótesis de trabajo de esta tesis doctoral consiste en demostrar que los algoritmos de CN ofrecen una herramienta potente para la resolución de los problemas de diseño de VANETs introducidos anteriormente. Así, se plantean tres ejes básicos:

- **Optimización *off-line* de protocolos**: Uso de técnicas CN para la configuración eficiente de parámetros de protocolos de transferencia de archivos y encaminamiento de paquetes. En este estudio se han aplicado técnicas tanto mono-objetivo como multi-objetivo; a su vez también se han aplicado modelos paralelos de algunos de los algoritmos estudiados.

- **Optimización *on-line* de protocolos**: Como las VANETs operan de forma distribuida, se plantea la utilización de estrategias CN distribuidas y dinámicas para el diseño de nuevos protocolos tanto de difusión de mensajes como de selección de tecnología de acceso al medio.
- **Diseño eficiente de la infraestructura**: Este problema puede modelarse de forma natural como multi-objetivo (maximizar calidad de servicio proporcionada y minimizar coste de despliegue), así que se va a aplicar un algoritmo multi-objetivo paralelo para tratar este problema de forma eficiente.

Cabe destacar que los problemas se han resuelto sobre instancias realistas definidas utilizando información real de la ciudad de Málaga.

C.5 Optimización *Off-line* de Protocolos

La optimización de protocolos *off-line* consiste en la búsqueda automática de configuraciones factibles de los parámetros que gobiernan los protocolos para optimizar su rendimiento. Éste no es un problema fácil. El número y el rango de los parámetros que gobiernan el protocolo definen un espacio de búsqueda lo suficientemente grande y desconocido para hacer inútil el uso de métodos exactos y enumerativos para resolverlo. Por contra, el uso de CN es viable puesto que calculan configuraciones optimizadas en tiempos de ejecución razonables.

La estrategia de optimización seguida en esta tesis ha consistido en aplicar de forma conjunta un algoritmo CN y un proceso de simulación de una VANET que informe de la calidad de las soluciones tentativas que se van creando. Así, las diferentes soluciones (configuraciones del protocolo que se optimiza) que calculan los algoritmos de optimización son evaluadas por un simulador que configura los nodos con la solución a evaluar. Tras la simulación, se analizan distintas métricas sobre las comunicaciones, y con ellas se evalúa la calidad o *fitness* de la solución (ver Figura C.2).

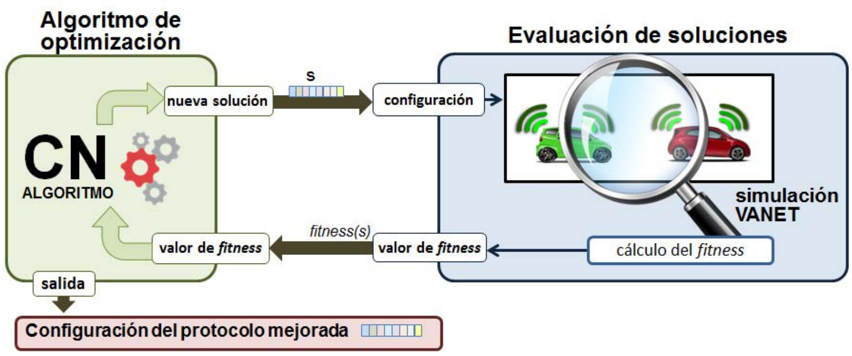


FIGURA C.2: Estrategia de optimización *off-line* de protocolos VANET.

Esta tesis ha tratado la optimización de dos tipos de protocolos: el protocolo de transferencia de archivos VDTP y los protocolos de encaminamiento de paquetes OLSR y AODV. Al ser protocolos de tipo distinto se han utilizado distintos tipos de métricas para evaluar la calidad de las soluciones. El primer tipo de protocolo se ha evaluado en términos de cantidad de datos transmitidos (Kbytes), número de segmentos de datos perdidos, tiempo de transmisión (segundos) y tasa de transferencia (Kbytes/segundo). Los protocolos de encaminamiento se han evaluado en función del porcentaje de paquetes recibidos respecto de los enviados o PDR, el tiempo que emplea un paquete en llegar a su destino o E2ED (milisegundos), la tasa de carga generada por un protocolo en relación con la cantidad de datos enviados o NRL.

Todos los problemas de optimización *off-line* se han resuelto sobre distintas instancias VANET definidas sobre la ciudad de Málaga. A su vez, cabe destacar que los protocolos mejorados se han enfrentado a una validación exhaustiva. Ésta consiste en una comparación con diferentes versiones en el estado del arte del mismo protocolo sobre diversos escenarios VANET que representan una multitud de situaciones distintas.

C.5.1 Optimización de la Transferencia de Archivos

El primer problema abordado es la optimización de la transferencia de archivos en VANET empleando el protocolo VDTP, denominado problema FTC (García-Nieto y col., 2010). Este problema consiste en encontrar la configuración de tres los parámetros que gobiernan el protocolo para maximizar la cantidad de datos enviados y minimizar tanto el tiempo de transmisión como el número de segmentos perdidos. Se ha estudiado el problema en entornos urbanos e interurbanos (autopista).

El uso de técnicas exactas para resolver este problema es impracticable debido al tamaño del espacio de búsqueda y al tiempo elevado de evaluación de las soluciones (simulaciones VANET). Así pues, hemos tratado el problemas utilizando un conjunto de técnicas CN: PSO, DE, GA, ES y SA. Éstas han empleado el simulador ns-2 para evaluar las soluciones. El *fitness* que se va a minimizar se calcula evaluando 10 transferencias de archivos según la Ecuación C.1.

$$fitness(s) = \frac{1}{N}\sum_{i=1}^{N}\frac{tiempo\_transmision_i + paquetes\_perdidos_i}{\log(datos\_transferidos_i + K)} \tag{C.1}$$

Evaluación Experimental

Para la evaluación experimental se realizado 30 ejecuciones independientes de cada algoritmo, cada una ejecutando 1000 evaluaciones de *fitness*. Los algoritmos poblacionales como PSO, DE, GA y ES operan sobre 20 soluciones (partículas o individuos), por lo que llevan a cabo 50 iteraciones. El SA, al ser un algoritmo de trayectoria, lleva a cabo 1000 iteraciones.

Las soluciones se han evaluado sobre dos escenarios, uno urbano y otro interurbano (en función de dónde se ha resuelto el problema), en los que hay una VANET de 30 vehículos de los cuales 20 realizan transferencias de archivos.

Los resultados obtenidos en entornos urbanos muestran que PSO obtiene los mejores valores de *fitness* (media, mediana y test estadístico de Friedman), seguido de DE y GA. Para entornos interurbanos los resultados varían algo más, puesto que el algoritmo que presenta mejor media de *fitness* es PSO y mejor mediana es SA. Además, según los test estadísticos no existen diferencias significativas entre PSO, SA y GA en el caso interurbano.

Validación de los Resultados

Comparando las soluciones obtenidas por los algoritmos con la propuesta estándar propuesta para VDTP, se comprueba que las configuraciones optimizadas mejoran el estándar en casi todos los casos. Concretamente, la configuración obtenida por PSO es la que ofrece un rendimiento más competitivo para las dos instancias estudiadas. Si se compara ésta con la estándar se ha observado que PSO mejora entre un 24 % y un 36 % la tasa de datos efectiva.

C.5.2 Encaminamiento Eficiente de Paquetes con OLSR

El problema de optimización *off-line* de OLSR consiste en buscar la configuración de los ocho parámetros principales del protocolo para optimizar su rendimiento en VANETs (maximizar PDR y minimizar E2ED y NRL). Siete parámetros son reales y uno entero. De nuevo las técnicas CN pueden calcular buenas soluciones en tiempos de ejecución aceptables.

Así pues, hemos analizado la optimización de OLSR utilizando un conjunto te técnicas CN: PSO, DE, GA y SA. A su vez, también hemos incluido una búsqueda aleatoria (RAND) como algoritmo base. El rendimiento de OLSR se evalúa en términos de PDR, NRL y E2ED. Así, cuando una solución tenga que ser evaluada se simulará usando ns-2 que devolverá los valores resultantes tras medir las tres métricas. Empleando dichos valores se evaluará la función objetivo que se va a minimizar y que se calcula según la siguiente ecuación (Toutouh y col., 2012b):

$$fitness(s) = w_2 \cdot NRL(s) + w_3 \cdot E2ED(s) - w_1 \cdot PDR(s) \tag{C.2}$$

Evaluación Experimental

Para la evaluación experimental se han realizado 30 ejecuciones independientes de cada uno de los algoritmos. Estos utilizan como criterio de parada evaluar 1000 veces la función objetivo. Los algoritmos poblacionales (PSO, DE y GA) se han configurado con 10 individuos, por lo que llevan a cabo 100 pasos iterativos (100×10=1000 evaluaciones de fitness).

Las soluciones calculadas se evalúan en una simulación donde 30 nodos de una VANET urbana intercambian paquetes encaminando los paquetes utilizando el protocolo OLSR configurado según la solución que se evalúe en cada momento.

Según los resultados del proceso de optimización, SA obtiene los mejores resultados en función de la media y la mediana. DE, PSO y GA son segundo, tercer y cuarto, respectivamente. Como era de esperar, RAND obtiene los peores resultados. Estos resultados se confirman de forma estadística puesto que el *ranking* del test de Friedman los ordena de este mismo modo (SA, DE, PSO, GA y RAND).

Validación de los Resultados

Los resultados se han validado primer, sobre el escenario utilizado para la optimización, y segundo sobre un conjunto de 54 escenarios distintos que presentan diferentes tamaños, densidades de tráfico vehicular y aplicaciones VANET.

Sobre el escenario de la optimización, los protocolos configurados utilizando CN obtienen los mejores resultados en comparación con el estándar de OLSR RFC 3626 (Clausen y Jacquet, 2003) y otras tres configuraciones de la literatura (Gómez y col., 2005).

Se ha hecho posteriormente una validación de la configuración aprendida sobre 54 escenarios nunca vistos para evaluar su robustez. Para la validación sobre los 54 escenarios distintos se ha comparado las configuraciones optimizadas con el estándar. En este caso se puede observar la fortaleza de las configuraciones optimizadas, puesto que reducen más de un 50 % de media la carga generada (uno de los principales problemas de este protocolo). Por lo general siempre se han reducido los tiempos de entrega de paquetes. Todo ello con una rebaja de la tasa de entrega de paquetes insignificante menor al 5 %.

C.5.3 Reducción del Consumo Energético de OLSR

La necesidad de tener la información continuamente actualizada sobre las rutas que tiene OLSR provoca otro de sus mayores problemas: el consumo elevado de energía por parte de los nodos. En las VANETs los vehículos no tienen unas restricciones de energía elevadas. En cambio otros nodos como sensores, paneles, estaciones base, etc. a veces solo se pueden abastecer de la energía de baterías limitadas o paneles solares. Por ello, en esta tesis también hemos tratado la optimización de la eficiencia energética de los protocolos de comunicación.

Sin embargo, una reducción excesiva del consumo energético de los protocolos puede llevar a una bajada del rendimiento o a un mal funcionamiento de los mismos. Por eso es importante tener en cuenta el rendimiento cuando se optimiza la energía. En el trabajo que aquí se presenta se ha minimizado la energía que consume OLSR, pero se ha empleado el PDR para garantizar la calidad de servicio. En este caso se va a permitir una pérdida de hasta un 15 % del PDR sobre el que obtendría el estándar (Toutouh y col., 2013).

Así pues, el problema de optimización consiste en encontrar la configuración que minimice la energía consumida por OLSR sin conllevar una disminución de más del 15 % del PDR. En este caso el problema lo hemos tratado aplicando una algoritmo evolutivo paralelo (modelo maestro-esclavo) basado en GA (pGA). La evaluación de la función objetivo (ver Ecuación C.3) se ha paralelizado para que el algoritmo pueda computarla más veces sin conllevar tiempos de cómputo prohibitivos, puesto que ésta se realiza simulando una VANET urbana con ns-2.

$$fitness(s) = \begin{cases} fitness_Q(s) & \text{si } PDR \geq 0,85 \times PDR \\ fitness_P(s) & \text{si } PDR < 0,85 \times PDR \end{cases} \tag{C.3}$$

La función de *fitness* que se minimiza se presenta en la Ecuación C.3. Cuando la reducción del PDR que genera la reducción de la energía es menor al 15 % se evalúa la solución según la Ecuación C.4. Se la merma del PDR es mayor, se aplica una penalización al *fitness* (ver Ecuación C.5) para que tenga la opción de mantenerse en la población.

$$fitness_Q(s) = \Delta + \left(\omega_1 \cdot \frac{E(s)}{E_{RFC}} + \omega_2 \cdot \frac{PDR(s)}{PDR_{MAX}}\right) \tag{C.4}$$

$$fitness_P(s) = fitness_Q(s) + \left((0{,}85 \cdot PDR_{RFC} - PDR(s)) \cdot \frac{E(s)}{E_{RFC}}\right) \tag{C.5}$$

Evaluación Experimental

Para analizar la eficacia del pGA en la resolución de este problema, se han probado tres implementaciones distintas: pGA-8, pGA-16 y pGA-24, que difieren en el número de individuos (o hebras) que emplean. Todas llevan a cabo 100 generaciones en cada ejecución. La evaluación experimental comprende 30 ejecuciones independientes de cada una de las implementaciones.

Como era de esperar, pGA-24 ha obtenido los mejores resultados y es el más competido según el test estadístico de Kruskal-Wallis. En el dominio del problema, la mejor configuración obtenida ha reducido en más de un 30 % la energía consumida y la degradación del PDR ha sido menor al 12 %, bien dentro de nuestras restricciones.

Cuando se emplean algoritmos paralelos es importante estudiar la eficiencia computacional de los mismos. En este caso los resultados han sido satisfactorios porque de media la eficiencia computacional ha sido mayor al 70 % en todos los caso (72 %, 74 % y 80 % para pGA-8, pGA16 y pGA-24, respectivamente).

Validación de Resultados

Los experimentos de validación se han llevado a cabo comparando las mejores configuraciones encontradas por los tres pGA con la versión estándar del OLSR. Para ello se han simulado 36 VANET urbanas que combinan distintos tamaños, densidades de tráfico y aplicaciones.

Los resultados de estos experimentos confirman que las configuraciones calculadas por el pGA consiguen que los nodos consuman menos energía. Concretamente la configuración devuelta por el pGA-24 es la más eficiente (la mejor según el test de Friedman), proporcionando un ahorro medio de un 40 % de energía con una reducción menor al 8 % del PDR respecto al estándar. Todo esto lo consigue mejorando en otras métricas como el E2ED y el NRL.

C.5.4 Opimización Multi-Objetivo del Encaminamiento con AODV

AODV es un protocolo de encaminamiento *unicast* para redes inalámbricas *ad hoc* que pertenece a la familia de protocolos reactivos, es decir, calcula la ruta cuando se va a empezar a transmitir y se mantiene solo durante la transmisión. Se han propuesto diferentes variantes de AODV para VANETs porque al ser reactivo ofrece una tasa de entrega de datos aceptable y no genera una carga de control excesiva. Sin embargo, los tiempos de entrega se ven afectados por el proceso de búsqueda de la ruta.

En este trabajo hemos definido el problema de optimización multi-objetivo del protocolo AODV en VANETs (AODV MO-QoS) (Toutouh y Alba, 2015c). Buscamos configuraciones factibles de los 11 parámetros de AODV que optimicen dos objetivos enfocados a los dos requisitos más críticos en el encaminamiento de datos en VANETs: maximizar la cantidad de datos que se intercambian (PDR) y minimizar los tiempos de transmisión (E2ED). Estos dos objetivos son contrapuestos porque cuanto mayor es la cantidad de paquetes que viajan a través de la red, mayor es la probabilidad de que aparezcan problemas de acceso al medio por lo que los nodos tardan más en retransmitir los paquetes, y viceversa.

Por lo tanto se van a emplear dos algoritmos multi-objetivo (NSGA-II y SMPSO) para resolver el AODV MO-QoS (maximizar PDR y minimizar E2ED) en una VANET urbana. Tras los resultados de eficiencia computacional obtenidos en el estudio anterior (ver Sección C.5.3) estos algoritmos se han implementado para que paralelicen la evaluación de las funciones objetivo (simulaciones VANET con ns-2).

Evaluación Experimental

El problema se ha resuelto sobre un escenario VANET urbano en el que hay 30 vehículos intercambiando información. Para el análisis experimental se han implementado NSGA-II y SMPSO paralelos (pNSGA-II y pSMPSO) para que se ejecuten empleando poblaciones/cúmulos de 24 individuos/partículas (o hebras). Se han llevado a cabo 30 ejecuciones independientes de cada uno de los algoritmos. Como criterio de parada se ha definido el obtener un frente de Pareto de una calidad (hipervolumen) determinada.

Los frentes de Pareto devueltos por los dos algoritmos tienen un hipervolumen similar, puesto que se ha utilizado dicha métrica como criterio de parada. Sin embargo, el frente devuelto por pNSGA-II presenta estadísticamente una mayor diversidad y mejor valor de *epsilon* (convergencia).

En términos de eficiencia computacional, si bien ambos algoritmos presentan una eficiencia computacional bastante satisfactoria (90 % el pNSGA-II y 87 % el pSMPSO), los tiempos de ejecución del algoritmo basado en inteligencia de enjambre ha necesitado una media de un 35 % más de tiempo en terminar que el pNSGA-II.

Validación de Resultados

La validación de resultados en este caso es más que interesante, puesto que en la literatura ya había estudios en los que se optimizaba AODV utilizando CN con los que comparar nuestra nueva metodología (formulación multi-objetivo y uso de algoritmos paralelos). Así pues, se han comparado una configuración calculada por cada uno de los algoritmos que aquí se han analizado (aquellas que minimiza la distancia con el vector ideal) con una calculada aplicando PSO propuesta por García-Nieto y Alba (2010), otra utilizando una versión paralela del mismo algoritmo (pPSO) presentada en Toutouh y Alba (2012c) y la configuración estándar (Perkins y col., 2003). Para ello se han evaluado las comunicaciones entre los nodos de las VANETs de todas estas configuraciones en 30 escenarios urbanos distintos (tamaños, densidad de tráfico y aplicaciones).

Los resultados han mostrado que las configuraciones devueltas por los algoritmos multi-objetivo obtienen los mejores resultados en términos de PDR. Las configuraciones obtenidas por PSO y pPSO han obtenido mejores resultados en términos de E2ED, pero con una penalización acusada en el PDR. Si tenemos en cuenta las configuraciones obtenidas por pNSGA-II y pSMPSO, la segunda es la obtiene mejores resultados en términos de E2ED.

Cabe destacar que estos resultados serían distintos si se hubiesen escogido otras soluciones de los frentes de Pareto calculados. Esta es una de las grandes ventajas de utilizar optimización multi-objetivo: se puede escoger una solución distinta (ya calculada a la vez que las demás) en función de cuál es la situación actual de la red y de los requisitos de la aplicación.

C.6 Optimización *On-line* de Protocolos

La difusión de mensajes o *beacons* es fundamental para muchas de las aplicaciones VANET cooperativas para la seguridad vial o CVS. Los vehículos tienen que estar constantemente difundiendo dichos mensajes con una frecuencia determinada para informar a los nodos vecinos sobre su posición, velocidad, etc. y para controlar otros aspectos de las comunicaciones.

La frecuencia a la que se difunden los mensajes determina la resolución de la información: cuanta mayor sea mejor informados están los vecinos. Sin embargo, cuando esta frecuencia no tiene en cuenta el tráfico de alrededor (número de vehículos o mensajes que circulan) puede llevar a la red a congestionarse. El rendimiento de los protocolos de difusión se puede medir en términos de carga de la red: cuanto mayor carga de la red sin llevar ésta a la congestión mejor será el algoritmo. También puede calcularse en términos del equilibrio en el uso de la red por cada uno de los nodos. Así pues, los métodos de difusión deben maximizar los dos objetivos (carga y equilibrio), sin llevar a la VANET a congestionarse.

En este contexto, en esta tesis se propone una nueva familia de algoritmos voraces distribuidos y dinámicos para la difusión de mensajes con adaptación de la frecuencia, los algoritmos *FREEDY*. Estos algoritmos se dividen en dos grupos en función de la fuente de información que utilizan: *Self FREEDY*, la información la obtienen de monitorizar la red, y *Swarm FREEDY*, que además de monitorizar la red obtienen información de los nodos de su vencindario. También se han dividido en función de qué tipo es la información que utilizan (la carga de red en función del número mensajes que recibe o el tamaño del vecindario en función de los emisores de los mensajes recibidos). Así que los algoritmos propuestos son: *Self o-FREEDY*, monitoriza la carga de la red; *Self n-FREEDY*, monitoriza el número de nodos vecinos; *Swarm o-FREEDY*, monitoriza e intercambia la carga de la red; y *Swarm n-FREEDY*, monitoriza e intercambia el tamaño de su vecindario (número de nodos en su rango de comunicaciones).

Evaluación Experimental

La evaluación experimental se ha realizado sobre nueve escenarios interurbanos en los que se ha analizado el rendimiento de los algoritmos propuestos simulando 100 veces cada uno. Además se han incluido en la evaluación dos algoritmos de difusión basados en ALOHA y en CSMA. El rendimiento se ha medido en términos de carga de la red y del equilibrio en el uso del canal.

De acuerdo con los resultados obtenidos, los algoritmos Swarm FREEDY son más competitivos que los Self FREEDY. Por esto, podemos decir que el intercambio de información entre los nodos mejora la percepción real del estado de la red y ésta se puede usar de forma más eficiente. Concretamtente, Swarm o-FREEDY, que utiliza la información de la carga de la red como medida para autoconfigurarse, es el algoritmo que mejores resultados ha obtenido (valores de compromiso entre las dos métricas que optimizar). Finalmente, los algoritmos Self FREEDY se comportaron de forma más eficiente que los métodos de difusión base (Aloha y CSMA).

C.7 Diseño Eficiente de la Infraestructura

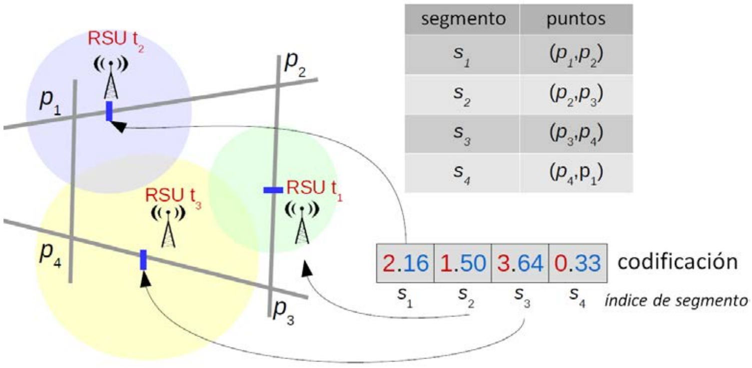
Las estaciones base o RSUs son parte fundamental en las VANETs por los servicios que dan. Disponer de una infraestructura instalada en las ciudades mejoraría las capacidades de comunicación de los vehículos y se podrían implantar nuevos servicios basados en comunicaciones V2I. El problema RSU-DP consiste en determinar la cantidad de RSUs que se va a desplegar, así como las localizaciones en las que se va a colocar cada una de ellas. Una RSU (antena) se puede colocar en un segmento que representa una carretera o una sección de la misma.

Los objetivos perseguidos son obtener la mayor calidad de servicio (en función del número de vehículos a los que se les da servicio y el tiempo del mismo) y reducir los costes de instalación. En esta tesis se ha formulado por primera vez en la literatura como problema con dos objetivos diferentes de forma explícita (multi-objetivo): maximizar cobertura y minimizar coste. El problema se ha tratado utilizando una versión paralela de NSGA-II al que se le han aplicado operadores específicos de inicialización y mutación. Más información sobre dichos operadores está en nuestro estudio presentado en Massobrio y col. (2015b).

En este trabajo se ha tratado este problema empleando información real sobre el tráfico publicada por el Ayuntamiento de Málaga. El área utilizada para definir la instancia cubre 42,55 km^2 de la ciudad, incluye 106 puntos para definir 121 segmentos con longitudes de entre 55 m y 1556 m. Además se han utilizado especificaciones reales sobre antenas que cumplen con el estándar IEEE 802.11p (para redes vehiculares) que se encuentran en el mercado (información de potencia/cobertura y precios).

C.7.1 Formulación del Problema

Las **soluciones se representan** como vectores de tamaño n del tipo real. En cada posición i del vector se almacena la información sobre la RSU instalada en el segmento i de la instancia: la parte entera del número real indica el tipo de RSU (0 significa que no hay RSU) y la parte real del número indica la sección del segmento en la que se ubica la RSU y está en el rango [0, 1). En el ejemplo de la Figura C.3, el valor *1,50* en la posición 2 del vector (s_2) indica que en el segmento 2 que va desde el punto p_2 al p_3 hay ubicada en el centro del segmento (1,*50*) una RSU con la antena del tipo 1 (*1*,50).



segmento	puntos
s_1	(p_1,p_2)
s_2	(p_2,p_3)
s_3	(p_3,p_4)
s_4	(p_4,p_1)

FIGURA C.3: Representación de soluciones en el RSU-DP.

El **cálculo de los dos objetivos** se hace de la siguiente forma. El coste total se obtiene de sumar el precio total de las antenas (según el tipo) que se han colocado en el escenario. Para el cálculo de la calidad de servicio se consideran las distancias y valores como se muestran en la Figura C.4: la RSU localizada en el punto $\times$ cubre los subsegmentos c_1 (en el segmento s_1), c_2 (en s_2), c_3 (en s_3) y c_4 (en s_4). El número efectivo de coches atendidos es $\sum_{i=1}^{i=4} NV(s_i) \times \frac{c_i}{sp(s_i)}$. Donde $NV(s_i)$ es el número total de vehículos que circulan por s_i.

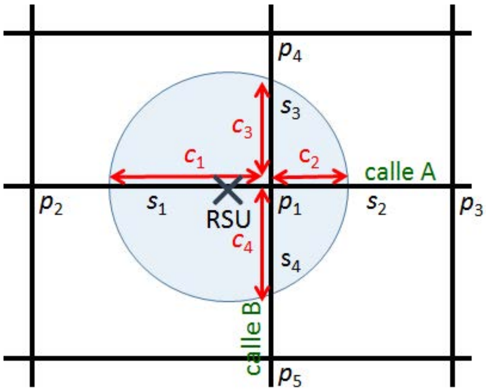


FIGURA C.4: Cálculo de la calidad de servicio que proporciona una RSU (RSU-DP).

C.7.2 Evaluación Experimental

El problema se ha tratado sobre la instancia que se ha mencionado antes (a la que llamamos *normal*), pero también se han definido dos instancias de forma probabilística: *bajo*, que tiene reducida la densidad de tráfico hasta un 20 % para cada segmento, y *alto*, que tiene incrementada la densidad hasta un 20 %.

Se ha ejecutado de forma independiente 30 veces el algoritmo para cada una de las instancias. A su vez, se han comparado los resultados con dos algoritmos voraces empleados en la literatura actual para tratar este mismo problema (uno que optimiza la calidad de servicio y otro que minimiza el coste).

Los resultados indican que el algoritmo de CN empleado mejora al algoritmo voraz que maximiza la calidad de servicio en un 6 % necesitando el mismo coste. Cuando se compara el CN con el voraz que minimiza el coste, nuestra propuesta mejora el coste en un 37 % el coste mientras obtiene la misma calidad de servicio (este resultado significa un ahorro de 5218,4 $ en una inversión de 14079,7 $). La Figura C.5 muestra una de las soluciones obtenidas por el algoritmo de CN.

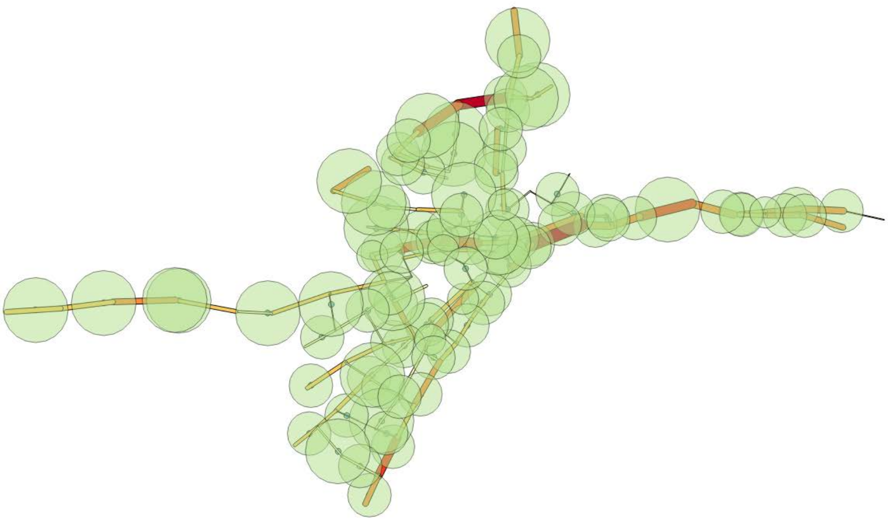


FIGURA C.5: Representación sobre el mapa de una solución obtenida al problema RSU-DP.

C.8 Pruebas Reales de Concepto

Para poder evaluar la validez de los resultados obtenidos mediante simulación sobre la optimización de protocolos, se ha llevado a cabo un estudio experimental empleando vehículos y dispositivos reales en las carreteras abiertas al tráfico de Málaga (Toutouh y Alba, 2011c). A su vez, se ha hecho otro estudio en el que se analizado la posibilidad de poder utilizar dispositivos móviles para las comunicaciones V2V (Toutouh y Alba, 2016).

C.8.1 Protocolos Optimizados en Entornos Reales

Para comprobar la eficiencia de los protocolos optimizados, se ha analizado el uso del VDTP estándar y el mejorado con CN (ver Sección C.5.1) en la transferencia de archivos. Para ello se han instalado dos ordenadores portátiles en sendos vehículos y se han realizado la transferencia de archivos de distintos tamaños (100 KBytes, 500 KBytes, 1 MByte, 5 MBytes y 10 MBytes). A su vez se han evaluado las transferencias de archivos a distintas velocidades: primero, velocidades comprendidas entre 20 y 30 km/h, y segundo, de entre 40 y 50 km/h.

Los resultados muestran que: **a)** los protocolos optimizados ofrecen una mayor velocidad de transferencia de datos que el estándar, **b)** a mayor tamaño del archivo que se transfiere, mayor es la tasa de transferencia de datos, y **c)** la velocidad afecta negativamente a la calidad de las comunicaciones (una mayor velocidad de los vehículos conlleva una menor tasa de transferencia).

C.8.2 Dispositivos Móviles en Comunicaciones Vehiculares

Este estudio ha consistido en evaluar las capacidades de distintos dispositivos móviles para su uso en VANETs. Concretamente se han analizado teléfonos móviles inteligentes, tabletas y ordenadores portátiles, que están fácilmente disponibles en el mercado y que muchos usuarios no especializados tienen. Para ello se han equipado pares de vehículos con dichos dispositivos y se han analizado las comunicaciones utilizando las interfaces Wi-Fi (IEEE 802.11g) todos con todos.

Se han realizado numerosas pruebas en las que se han probado las comunicaciones entre dos vehículos que se mueven a velocidades que van desde los 15 km/h a los 50 km/h por carreteras abiertas al tráfico. A su vez, se han estudiado cómo se ven afectadas las comunicaciones en función de la distancia a la que se encuentran los nodos. Así, se han hecho distintos experimentos modificando la distancia (más o menos constantes) entre los vehículos (se han probado distancias de 25, 50, 75, 100, 125 y 150 metros). Los vehículos intercambiaban flujos de 100 paquetes de datos de distintos tamaños (32, 64, 128, 256 y 512 bytes).

El análisis de los resultados obtenidos nos permite observar que los teléfonos inteligentes posibilitan el intercambio de paquetes de datos de texto con nodos alejados hasta 75 metros. Los nodos equipados con tabletas pueden comunicaciones con nodos a una distancia de 125 metros. Por último, los ordenadores portátiles son capaces de intercambiar información multimedia con vehículos alejados a distancias mayores de 150 metros. Además, debido a la mayor sensibilidad de la interfaz Wi-Fi de los portátiles, si un portátil se comunica con un teléfono o una tableta, estos pueden comunicarse con el portátil aunque esté a mayor distancia de las que se han presentado antes (más de 75 m si es un teléfono móvil y más de 125 m si es una tableta).

C.9 Conclusiones y Trabajo Futuro

En esta tesis doctoral hemos abordado la resolución de varios de los principales problemas en el ámbito de las VANET: la optimización *off-line* de protocolos de transferencia de archivos y de encaminamiento de paquetes; la optimización *on-line* de protocolos de difusión de mensajes; y el diseño eficiente de la infraestructura de estaciones base. Todos ellos se han tratado aplicando CN.

Se ha observado que la optimización *off-line* de protocolos es una herramienta que ofrece unos resultados satisfactorios tanto a la hora de mejorar las comunicaciones como reducir el consumo de recursos de los protocolos. Se ha conseguido mayores tasas de transferencia cuando se ha optimizado VDTP. Facilitando que se puedan enviar archivos de datos de mayor tamaño. Los protocolos de encaminamiento han mejorado su escalabilidad (reduciendo la cantidad de tráfico de control que generan), a la vez que han mejorado o mantenido su eficiencia (cantidad de paquetes entregados y tiempos de envío). Cabe destacar que esta metodología de optimización se puede aplicar a cualquier protocolo VANET, no solo sobre los que se han analizado en esta tesis.

Respecto al comportamiento de los algoritmos CN y la estrategia de resolución de los problemas de optimización *off-line* de protocolos, se ha observado que los algoritmos paralelos ofrecen una eficiencia computacional satisfactoria (ésta ha ido desde el 70 % al 90 %). Además, las parametrizaciones que ofrece la formulación multi-objetivo han mejorado las mismas obtenidas por algoritmos mono-objetivo.

El problema de optimización de protocolo *on-line* consiste en diseñar algoritmos basados en CN que se ejecuten de forma eficiente durante el funcionamiento de la red. En esta tesis se han diseñado la familia de protocolos *FREEDY* para la difusión de mensajes de forma eficiente. Estos algoritmos dinámicos distribuidos utilizan información de la carga del canal o el número de nodos vecinos para adaptar de forma eficiente la frecuencia con la que se difunden los mensajes. De los algoritmos propuestos el *Swarm o-FREEDY* (los nodos utilizan la información que monitorean y la que intercambian con el vecindario para el cálculo de las nuevas frecuencias de envío de mensajes) es el que presenta un uso más eficiente del canal a la vez que balancea mejor el acceso al medio por parte de los nodos.

El último problema de optimización analizados, es el del diseño inteligente de la infraestructura de estaciones base. En este caso se ha propuesto de forma novedosa una formulación multi-objetivo del problema y se ha resuelto utilizando una versión paralela de NSGA-II. Los resultados obtenidos sobre una instancia realista de Málaga muestran que el algoritmo CN mejora el resultado de distintas heurísticas propuestas en la literatura, tanto en calidad de servicio con mismo coste como en reducción del coste con la misma calidad de servicio.

A su vez, el trabajo realizado en esta tesis se ha completado con pruebas de concepto utilizando vehículos y dispositivos reales. Con los resultados obtenidos se puede observar que los protocolos optimizados efectivamente ofrecen un mejor rendimiento que la versión estándar. Esto se ha estudiado sobre el protocolo VDTP de transferencia de archivos. Asimismo, se ha observado que se pueden utilizad dispositivos móviles personales para realizar comunicaciones VANET (IEEE 802.11g), aunque según la naturaleza de los mismos se disfruta de una calidad de servicio u otra. Los teléfonos inteligentes, las tabletas y los ordenadores portátiles proporcionan rangos de cobertura de hasta 75 metros, hasta 125 metros y de más de 150 metros, respectivamente.

Como evaluación general de la tesis, los resultados obtenidos en los distintos problemas analizados presentan a la CN como una herramienta prometedora para resolver de forma eficiente problemas en redes vehiculares.

Las líneas de trabajo futuro avanzarán en la mejora de las técnicas de optimización utilizando CN. Por ejemplo, la optimización *off-line* podría emplear simulación Monte-Carlo para evaluar las soluciones en distintos escenarios a en paralelo. Así, se podría mejorar la robustez de las parametrizaciones optimizadas que se han calculado. En la optimización *on-line* de protocolos, se podría aplicar técnicas modernas basadas en inteligencia de enjambre (tipo ACO o ABC) para mejorar los algoritmos propuestos. A su vez, el problema RSU-DP se podría analizar sobre instancias más grandes y de otras ciudades, incluyendo nueva información relevante sobre el tráfico (como por ejemplo, accidentes o puntos de interés). Finalmente, se plantea la definición de experimentos reales empleando un mayor número de vehículos y dispositivos con interfaces de red IEEE 802.11p, tan pronto como se extienda su oferta en el mercado.

List of Figures

List of Tables

Index

References

3GPP (2010). *Overview of 3GPP Release 8*. [online] Available in http://www.3gpp.org/Release-8. Retrieved in July 2015.

Abdou, W., A. Henriet, C. Bloch, D. Dhoutaut, D. Charlet, and F. Spies (2011). "Using an evolutionary algorithm to optimize the broadcasting methods in mobile ad hoc networks". In: *Journal of Network and Computer Applications* 34.6. Control and Optimization over Wireless Networks, pp. 1794 –1804.

Al-Khusaibi, H., F. Al-Wardi, F. Sawalha, and W. Xiang (2006). "Experiment and Analysis on the Comparison of the IEEE 802.11a and 802.11g Wireless Local Area Networks". In: *Electro/information Technology, 2006 IEEE International Conference on*, pp. 13–16.

Al-Sultan, S., M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan (2014). "A comprehensive survey on vehicular Ad Hoc network". In: *Journal of Network and Computer Applications* 37, pp. 380–392.

Alba, E. (2005). *Parallel Metaheuristics: A New Class of Algorithms*. Wiley-Interscience.

Alba, E., B. Dorronsoro, F. Luna, and P. Bouvry (2005). "A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs". In: *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pp. 1–8.

Alba, E., F. Almeida, M. Blesa, C. Cotta, M. Diaz, I. Dorta, J. Gabarró, J. González, C. León, L. Moreno, J. Petit, J. Roda, A. Rojas, and F. Xhafa (2006). "MALLBA: A library of skeletons for combinatorial optimisation". In: *Parallel Computing* 32.5-6, pp. 415–440.

Alba, E., B. Dorronsoro, F. Luna, A. Nebro, P. Bouvry, and L. Hogie (2007a). "A cellular MOGA for optimal broadcasting strategy in metropolitan MANETs". In: *Computer Communications* 30.4, pp. 685–697.

Alba, E., A. Cervantes, J. Gómez, P. Isasi, M. Jaraíz, C. León, C. Luque, F. Luna, G. Miranda, A. Nebro, R. Pérez, and C. Segura (2007b). "Metaheuristic Approaches for Optimal Broadcasting Design in Metropolitan MANETs". English. In: *Computer Aided Systems Theory – EUROCAST 2007*. Ed. by R. Moreno Díaz, F. Pichler, and A. Quesada Arencibia. Vol. 4739. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 755–763.

Alba, E. and M. Tomassini (2002). "Parallelism and evolutionary algorithms". In: *Evolutionary Computation, IEEE Transactions on* 6.5, pp. 443–462.

Alba, E., S. Luna, and J. Toutouh (2008). "Accuracy and Efficiency in Simulating VANETs". In: *Modelling, Computation and Optimization in Information Systems and Management Sciences, Second International Conference (MCO)*. Vol. 14. Communications in Computer and Information Science. Metz, France - Luxembourg: Springer, pp. 568–578.

ALFA (2015). *Alfa Network Inc.* [online] Available in http://www.alfa.com.tw/. Retrieved October 2015.

Artimy, M. M., W. Robertson, and W. J. Phillips (2005). “Assignment of Dynamic Transmission Range Based on Estimation of Vehicle Density”. In: *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*. VANET ’05. Cologne, Germany: ACM, pp. 40–48.

Aslam, B., F. Amjad, and C. Zou (2012). “Optimal roadside units placement in urban areas for vehicular networks”. In: *IEEE Symposium on Computers and Communications*, pp. 423–429.

ASTM (2003). *ASTM e2213 - 03(2010) standard specification for telecommunications and in-formation exchange between roadside and vehicle systems — 5 Ghz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications*. [online] Available in http://www.astm.org/Standards/E2213.htm. Retrieved October 2012.

Ben Brahim, M., W. Drira, and F. Filali (2014). “Roadside units placement within city-scaled area in vehicular ad-hoc networks”. In: *Connected Vehicles and Expo (ICCVE), 2014 International Conference on*, pp. 1010–1016.

Benslimane, A., T. Taleb, and R. Sivaraj (2011). “Dynamic Clustering-Based Adaptive Mobile Gateway Management in Integrated VANET - 3G Heterogeneous Wireless Networks”. In: *Selected Areas in Communications, IEEE Journal on* 29.3, pp. 559–570.

Benslimane, A. (2004). “Optimized Dissemination of Alarm Messages in Vehicular Ad-Hoc Networks (VANET)”. English. In: *High Speed Networks and Multimedia Communications*. Ed. by Z. Mammeri and P. Lorenz. Vol. 3079. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 655–666.

Beyer, H.-G. and H.-P. Schwefel (2002). “Evolution strategies–A comprehensive introduction”. In: *Natural computing* 1.1, pp. 3–52.

Bitam, S., A. Mellouk, and S. Zeadally (2013). “HyBR: A Hybrid Bio-inspired Bee swarm Routing protocol for safety applications in Vehicular Ad hoc NETworks (VANETs)”. In: *Journal of Systems Architecture* 59.10, Part B. Advanced Smart Vehicular Communication System and Applications, pp. 953 –967.

— (2015). “Bio-Inspired Routing Algorithms Survey for Vehicular Ad Hoc Networks”. In: *Communications Surveys Tutorials, IEEE* 17.2, pp. 843–867.

Blum, C. and A. Roli (2003). “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”. In: *ACM Computing Surveys (CSUR)* 35.3, pp. 268–308.

Bonabeau, E., M. Dorigo, and G. Theraulaz (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford University Press, Inc.

Broadcom (2015). *Broadcom*. [online] Available in http://www.broadcom.com/. Retrieved October 2015.

Bychkovsky, V., B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden (2006). “A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks”. In: *12th ACM MOBICOM Conf.* Los Angeles, CA.

C. Perkins S. Ratliff, J. D. (2013). *Dynamic MANET On-demand (DYMO) Routing*. Tech. rep. draft-ietf-manet-dymo-26. [Online]. Available: https://tools.ietf.org/html/draft-ietf-manet-dymo-26. Fremont, CA, USA: IETF Secretariat.

C2C-CC (2007). *CAR 2 CAR Communication Consortium Manifesto.* Tech. rep. [Online]. Available: https://www.car-2-car.org/index.php?id=31. CAR 2 CAR Communication Consortium.

— (2015). *CAR 2 CAR Communication Consortium (C2C-CC)*. [online] Available in https://www.car-2-car.org. Retrieved October 2015.

CALM (2015). *Communications in Cooperative Intelligent Transport Systems - CALM for C-ITS*. [online] Available in http://calm.its-standards.info/. Retrieved October 2015.

CAMP (2005). *Vehicle Safety Communications Consortium (2005). Vehicle safety communications project task 3 final report: Identify intelligent vehicle applications enabled by DSRC*. Tech. rep. [Online]. Available: http://ntl.bts.gov/lib/29000/29500/29505/CAMP3scr.pdf. Washington, DC: CAMP.

Campolo, C., A. Molinaro, and R. Scopigno, eds. (2015). *Vehicular ad hoc Networks - Standards, Solutions, and Research*. Springer.
Cano, J. and P. Manzoni (2000). "A Performance Comparison of Energy Consumption for Mobile Ad Hoc Network Routing Protocols". In: *Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE Computer Society, pp. 57–64.
CARLINK (2006). *CARLINK Label CP3-005*. [online] Available in `http://carlink.lcc.uma.es/`. Retrieved October 2015.
Cavalcante, E., A. Aquino, G. Pappa, and A. Loureiro (2012). "Roadside Unit Deployment for Information Dissemination in a VANET: An Evolutionary Approach". In: *14^{th} Genetic and Evolutionary Computation Conference*, pp. 27–34.
CellCar (2013). *CellCar: Advanced Cellular Technologies for Connected Cars*. [online] Available in `http://www.cellcar.org/`. Retrieved October 2015.
Cetacea (2015). *Cetacea Wireless Solutions Company shop*. [online] Available in `https://shop.cetacea.com/`. Retrieved June 2015.
Chauhan, R. and A. Dahiya (2012). "AODV Extension using Ant Colony Optimization for Scalable Routing in VANETs". In: *Journal of Emerging Trends in Computing and Information Sciences* 3.2.
Chen, R., W.-L. Jin, and A. Regan (2010). "Broadcasting safety information in vehicular networks: issues and approaches". In: *Network, IEEE* 24.1, pp. 20–25.
Chen, W., R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu (2011). "A survey and challenges in routing and data dissemination in vehicular ad hoc networks". In: *Wireless Communications and Mobile Computing* 11.7, pp. 787–795.
Cheng, H., X. Fei, A. Boukerche, A. Mammeri, and M. Almulla (2013). "A Geometry-based Coverage Strategy over Urban VANETs". In: *10^{th} ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. Barcelona, Spain, pp. 121–128.
Cheng, H. and S. Yang (2010). "Genetic algorithms with immigrant schemes for dynamic multicast problems in mobile ad hoc networks". In: *Engineering Applications of Artificial Intelligence* 23 (5), pp. 806–819.
Chiong, R. (2009). *Nature-inspired algorithms for optimisation*. Vol. 193. Springer.
Clausen, T. and P. Jacquet (2003). *Optimized Link State Routing Protocol (OLSR)*. IETF RFC 3626, [online] Available in URL `http://www.ietf.org/rfc/rfc3626.txt`. United States.
Coello, C., G. Lamont, and D. Van Veldhuizen (2007). *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer-Verlag New York Inc.
CoMoSeF (2012). *CoMoSeF: Cooperative Mobility Services of the Future*. [online] Available in `http://www.list.lu/en/project/comosef/`. Retrieved October 2015.
Cooklev, T. (2004). *Wireless Communication Standards: A Study of IEEE 802.11, 802.15, 802.16*. IEEE Standards Association.
Correia, S., J. Celestino, and O. Cherkaoui (2011). "Mobility-aware Ant Colony Optimization routing for vehicular ad hoc networks". In: *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pp. 1125–1130.
Das, S. R., R. Castañeda, and J. Yan (2000). "Simulation-based Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks". In: *Mob. Netw. Appl.* 5.3, pp. 179–189.
De Castro, L. N. (2006). *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press.
De Rango, F. and M. Fotino (2009). "Energy efficient OLSR performance evaluation under energy aware metrics". In: *Proceedings of the 12th international conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems*. SPECTS'09. Istanbul, Turkey: IEEE Press, pp. 193–198.

De Rango, F., J.-C. Cano, M. Fotino, C. Calafate, P. Manzoni, and S. Marano (2008). “OLSR vs DSR: A comparative analysis of proactive and reactive mechanisms from an energetic point of view in wireless ad hoc networks”. In: *Computer Communications* 31.16, pp. 3843–3854.
Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons.
Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *Evolutionary Computation, IEEE Transactions on* 6.2, pp. 182–197.
Demšar, J. (2006). “Statistical comparisons of classifiers over multiple data sets”. In: *The Journal of Machine Learning Research* 7, pp. 1–30.
DGT (2015). *Dirección General de Tráfico*. [online] Available in http://www.dgt.es/es/el-trafico/recomendaciones/consejos-estacionales/index.shtml. Retrieved June 2015.
Dikaiakos, M. D., T. Nadeem, S. Iqbal, and L. Iftode (2005). “VITP: An Information Transfer Protocol for Vehicular Computing”. In: *in: VANET '05: Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks, ACM*. Press, pp. 30–39.
Ding, B., Z. Chen, Y. Wang, and H. Yu (2011). “An improved AODV routing protocol for VANETs”. In: *Int. Conf. on Wireless Communications and Signal Processing*. IEEE, pp. 1–5.
DIRICOM (2008). *Diseño Inteligente de Redes Inalámbricas de Comunicación (DIRICOM)*. [online] Available in http://diricom.lcc.uma.es/. Retrieved October 2015.
Djahel, S. and Y. Ghamri-Doudane (2012). “A robust congestion control scheme for fast and reliable dissemination of safety messages in VANETs”. In: *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pp. 2264–2269.
Dodig-Crnkovic, G. (2002). “Scientific methods in computer science”. In: *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia*, pp. 126–130.
Doefexi, A., S. Armour, B.-S. Lee, A. Nix, and D. Bull (2003). “An evaluation of the performance of IEEE 802.11a and 802.11g wireless local area networks in a corporate office environment”. In: *Communications, 2003. ICC '03. IEEE International Conference on*. Vol. 2, pp. 1196–1200.
Dorigo, M., V. Maniezzo, and A. Colorni (1996). “Ant system: optimization by a colony of cooperating agents”. In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 26.1, pp. 29–41.
Dorronsoro, B., P. Ruiz, G. Danoy, Y. Pigné, and P. Bouvry (2014). *Evolutionary algorithms for mobile ad hoc networks*. John Wiley & Sons.
Dressler, F. and O. B. Akan (2010). “A survey on bio-inspired networking”. In: *Computer Networks* 54.6, pp. 881–900.
Dua, A., N. Kumar, and S. Bawa (2014). “A systematic review on routing protocols for vehicular ad hoc networks”. In: *Vehicular Communications* 1.1, pp. 33–52.
Durillo, J. and A. Nebro (2011). “jMetal: A Java framework for multi-objective optimization”. In: *Advances in Engineering Software* 42, pp. 760–771.
Durillo, J., A. Nebro, F. Luna, and E. Alba (2008). “A study of master-slave approaches to parallelize NSGA-II”. In: *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pp. 1–8.
Dybowski, R., V Gant, P Weller, and R Chang (1996). “Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm”. In: *The Lancet* 347.9009, pp. 1146–1150.
Eberhart, R. C. and J. Kennedy (1995). “A new optimizer using particle swarm theory”. In: *Proceedings of the sixth international symposium on micro machine and human science*. Vol. 1. New York, NY, pp. 39–43.

ETSI (2010). "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments". In: *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51.

Fallah, Y. P., C. Huang, R. Sengupta, and H. Krishnan (2010). "Congestion control based on channel occupancy in vehicular broadcast networks". In: *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*. IEEE, pp. 1–5.

FCC (1999). *Federal Communications Commission news.* [online] Available in http://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html. Retrieved October 2015.

Festag, A., H. Fussler, H. Hartenstein, A. Sarma, and R. Schmitz (2004). "Fleetnet: Bringing Car-to-car Communication into the Real World". In: *Proceedings of the 11th World Congress on ITS*.

Fogel, L. J. (1999). *Intelligence through simulated evolution: forty years of evolutionary programming*. John Wiley & Sons.

Fogel, L. J., A. J. Owens, and M. J. Walsh (1966). *Artificial intelligence through simulated evolution*. John Wiley & Sons.

García-Nieto, J. and E. Alba (2010). "Automatic Parameter Tuning with Metaheuristics of the AODV Routing Protocol for Vehicular Ad-Hoc Networks". In: *EvoApplications (2)*. Vol. 6025. LNCS. Springer, pp. 21–30.

García-Nieto, J., J. Toutouh, and E. Alba (2010). "Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics". In: *Engineering Applications of Artificial Intelligence* 23.5. Advances in metaheuristics for hard optimization: new trends and case studies, pp. 795–805.

Gass, R., J. Scott, and C. Diot (2006). "Measurements of In-Motion 802.11 Networking". In: *7th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'06)*, pp. 69–74.

Gerla, M. and L. Kleinrock (2011). "Vehicular networks and the future of the mobile internet". In: *Computer Networks* 55.2, pp. 457–469.

Ghosh, A., D. R. Wolter, J. G. Andrews, and R. Chen (2005). "Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential". In: *Communications Magazine, IEEE* 43.2, pp. 129–136.

Glover, F. (1989). "Tabu search-part I". In: *ORSA Journal on computing* 1.3, pp. 190–206.

Glover, F. and G. A. Kochenberger (2003). *Handbook of metaheuristics*. Springer Science & Business Media.

Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley.

Gómez, C., D. García, and J. Paradells (2005). "Improving Performance of a Real Ad Hoc Network by Tuning OLSR Parameters". In: *ISCC '05: Proceedings of the 10th IEEE Symposium on Computers and Communications*. Washington, DC, USA: IEEE Computer Society, pp. 16–21.

Guerrero-Ibáñez, J., C. Flores-Cortés, and S. Zeadally (2013). "Vehicular Ad-hoc Networks (VANETs): Architecture, Protocols and Applications". English. In: *Next-Generation Wireless Technologies*. Ed. by N. Chilamkurti, S. Zeadally, and H. Chaouchi. Computer Communications and Networks. Springer London, pp. 49–70.

Haklay, M. M. and P. Weber (2008). "OpenStreetMap: User-Generated Street Maps". In: *IEEE Pervasive Computing* 7.4, pp. 12–18.

Hameed Mir, Z. and F. Filali (2014). "LTE and IEEE 802.11p for vehicular networking: a performance evaluation". In: *EURASIP Journal on Wireless Communications and Networking* 2014.1, p. 89.

Han, C., M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen (2012). "Analytical Study of the IEEE 802.11p MAC Sublayer in Vehicular Networks". In: *Intelligent Transportation Systems, IEEE Transactions on* 13.2, pp. 873–886.

Härri, J., F. Filali, and C. Bonnet (2006). "Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns". In: *Med-Hoc-Net 2006, 5th Annual Mediterranean Ad Hoc Networking Workshop*. Ed. by S. Basagni, A. Capone, L. Fratta, and G. Morabito. IFIP. Lipari, Sicily, IItaly.

Härri, J., M. Fiore, F. Filali, and C. Bonnet (2011). "Vehicular Mobility Simulation with VanetMobiSim". In: *Simulation* 87.4, pp. 275–300.

Hartenstein, H. and K. Laberteaux (2009). *VANET Vehicular Applications and Inter-Networking Technologies*. Intelligent Transport Systems. Upper Saddle River, NJ, USA: John Wiley & Sons.

Hogie, L., F. Guinand, and P. Bouvry (2004). "A heuristic for efficient broadcasting in the metropolitan ad hoc network". In: *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, pp. 727–733.

Holland, J. H. (1975). "Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence". In: *MIT Press*.

Hossain, E., G. Chow, V. C. Leung, R. D. McLeod, J. Mišić, V. W. Wong, and O. Yang (2010). "Vehicular telematics over heterogeneous wireless networks: A survey". In: *Computer Communications* 33.7, pp. 775–793.

HP (2015). *Hewlett-Packard*. [online] Available in http://www.hp.com/. Retrieved October 2015.

Huang, Y., S. Bhatti, and D. Parker (2006). "Tuning OLSR". In: *Proceedings of the IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, Helsiniki, Finland*, pp. 1–5.

Huhtonen, A. (2004). "Comparing AODV and OLSR routing protocols". In: *Telecommunications Software and Multimedia, 2004*, pp. 1–9.

Issariyakul, T. and E. Hossain (2008). *Introduction to Network Simulator NS2*. 1st ed. Springer Publishing Company, Incorporated.

Jogi, S. and M. Choudhary (2009). *Ultra Wideband Demystified Technologies, Applications, and System Design Considerations*. Vol. 4. River Publishers.

Karaboga, D. and B. Basturk (2007). "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm". In: *Journal of global optimization* 39.3, pp. 459–471.

Katsaros, K., R. Kernchen, M. Dianati, D. Rieck, and C. Zinoviou (2011). "Application of vehicular communications for improving the efficiency of traffic in urban areas". In: *Wireless Communications and Mobile Computing* 11.12, pp. 1657–1667.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). "Optimization by Simulated Annealing". In: *Science* 220.4598, pp. 671–680.

Knowles, J., L. Thiele, and E. Zitzler (2006). "A tutorial on the performance assessment of stochastic multiobjective optimizers". In: *Tik report* 214, pp. 327–332.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press.

Krajzewicz, D., J. Erdmann, M. Behrisch, and L. Bieker (2012). "Recent development and applications of SUMO–simulation of urban mobility". In: *International Journal On Advances in Systems and Measurements* 5.3&4.

Lai, P., X. Wang, N. Lu, and F. Liu (2009). "A reliable broadcast routing scheme based on mobility prediction for VANET". In: *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 1083–1087.

Langville, A. N. and C. D. Meyer (2011). *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press.

Lee, K., S.-H. Lee, R. Cheung, U. Lee, and M. Gerla (2007). "First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed". In: *2007 Mobile Networking for Vehicular Environments*, pp. 109–114.

Lee, K. C., U. Lee, and M. Gerla (2010). "Survey of routing protocols in vehicular ad hoc networks". In: *Advances in vehicular ad-hoc networks: Developments and challenges*, pp. 149–170.

Lee, U., J.-S. Park, S.-H. Lee, W. W. Ro, G. Pau, and M. Gerla (2008). "Efficient peer-to-peer file sharing using network coding in MANET". In: *Communications and Networks, Journal of* 10.4, pp. 422–429.

Li, G. and L. Boukhatem (2013). "Adaptive Vehicular Routing Protocol Based on Ant Colony Optimization". In: *Proceeding of the Tenth ACM International Workshop on Vehicular Internetworking, Systems, and Applications*. VANET '13. Taipei, Taiwan: ACM, pp. 95–98.

Li, W., L. Fu, B. Niu, S. Wu, and J. Wooley (2012). "Ultrafast clustering algorithms for metagenomic sequence analysis". In: *Briefings in bioinformatics*, bbs035.

Liang, W., B. Meng, X. He, and X. Zhang (2015). "GCM: A Greedy-Based Cross-Matching Algorithm for Identifying Users Across Multiple Online Social Networks". English. In: *Intelligence and Security Informatics*. Ed. by M. Chau, G. A. Wang, and H. Chen. Vol. 9074. Lecture Notes in Computer Science. Springer International Publishing, pp. 51–70.

Lin, W., M.-W. Li, K.-C. Lan, and C.-H. Hsu (2012). "A Comparison of 802.11a and 802.11p for V-to-I Communication: A Measurement Study". English. In: *Quality, Reliability, Security and Robustness in Heterogeneous Networks*. Ed. by X. Zhang and D. Qiao. Vol. 74. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, pp. 559–570.

Lin, X., R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen (2008). "Security in vehicular ad hoc networks". In: *Communications Magazine, IEEE* 46.4, pp. 88–95.

Lochert, C., B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve (2008). "Data Aggregation and Roadside Unit Placement for a Vanet Traffic Information System". In: *5^{th} ACM Int. Workshop on Vehicular Inter-Networking*. San Francisco, California, USA: ACM, pp. 58–65.

Lochert, C., B. Scheuermann, and M. Mauve (2007). "A survey on congestion control for mobile ad hoc networks". In: *Wireless Communications and Mobile Computing* 7.5, pp. 655–676.

Lownes, N. E. and R. B. Machemehl (2006). "VISSIM: A Multi-parameter Sensitivity Analysis". In: *Proceedings of the 38th Conference on Winter Simulation*. WSC '06. Monterey, California: Winter Simulation Conference, pp. 1406–1413.

Luna S. (2008). *VDTP: A File Transfer Protocol for Vehicular Ad hoc Networks*. Tech. rep. [Online]. Available: http://neo.lcc.uma.es/staff/jamal/downloads/VDTP.pdf. CAR-LINK.

Mange, D. and M. Tomassini (1998). *Bio-inspired computing machines: towards novel computational architectures*. PPUR presses polytechniques.

Mangel, T., M. Michl, O. Klemp, and H. Hartenstein (2011). "Real-World Measurements of Non-Line-Of-Sight Reception Quality for 5.9GHz IEEE 802.11p at Intersections". English. In: *Communication Technologies for Vehicles*. Ed. by T. Strang, A. Festag, A. Vinel, R. Mehmood, C. Rico Garcia, and M. Röckl. Vol. 6596. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 189–202.

Mangharam, R., D. Weller, R. Rajkumar, P. Mudalige, and F. Bai (2006). "GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks". In: *Mobile and Ubiquitous Systems - Workshops, 2006. 3rd Annual International Conference on*, pp. 1–8.

Martinez, F. J., C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni (2011). "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)". In: *Wireless Communications and Mobile Computing* 11.7, pp. 813–828.

Mashita, K. (2003). “Development of ASV in Japan -ASV Promotion Project, Phase-3 -”. In: *IATSS Research* 27.2, pp. 82 –87.

Massobrio, R., J. Toutouh, and S. Nesmachnow (2015a). “A Multiobjective Evolutionary Algorithm for Infrastructure Location in Vehicular Networks”. In: *7th European Symposium on Computational Intelligence and Mathematics (ESCIM 2015)*. Cádiz, Spain, pp. 1–6.

Massobrio, R., S. Bertinat, J. Toutouh, S. Nesmachnow, and E. Alba (2015b). “Smart placement of RSU for vehicular networks using multiobjective evolutionary algorithms”. In: *2nd LA-CCI Congress on Computational Intelligence (LA-CCI 2015)*. Curitiba, Brazil, pp. 1–6.

MAXCT (2014). *MAXCT: Imagine a new era in Smart Mobility*. [online] Available in http://roadme.lcc.uma.es/. Retrieved October 2015.

McCulloch, W. S. and W. Pitts (1943). “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Mejri, M. N., J. Ben-Othman, and M. Hamdi (2014). “Survey on VANET security challenges and possible cryptographic solutions”. In: *Vehicular Communications* 1.2, pp. 53 –66.

Menouar, H., M. Lenardi, and F. Filali (2007). “Movement Prediction-Based Routing (MOPR) Concept for Position-Based Routing in Vehicular Networks”. In: *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 2101–2105.

Mir, Z. H., J. Toutouh, F. Filali, and E. Alba (2015). “QoS-Aware Radio Access Technology (RAT) Selection in Hybrid Vehicular Networks”. English. In: *Communication Technologies for Vehicles*. Ed. by M. Kassab, M. Berbineau, A. Vinel, M. Jonsson, F. Garcia, and J. Soler. Vol. 9066. Lecture Notes in Computer Science. Springer International Publishing, pp. 117–128.

Mittag, J., F. Schmidt-Eisenlohr, M. Killat, J. Härri, and H. Hartenstein (2008). “Analysis and Design of Effective and Low-overhead Transmission Power Control for VANETs”. In: *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking*. VANET ’08. San Francisco, California, USA: ACM, pp. 39–48.

Molina-Gil, J., P. Caballero-Gil, and C. Caballero-Gil (2014). “Aggregation and probabilistic verification for data authentication in VANETs”. In: *Information Sciences* 262, pp. 172–189.

Morris, R., J. Jannotti, F. Kaashoek, J. Li, and D. Decouto (2000). “Carnet: A scalable ad hoc wireless network system”. In: *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*. ACM, pp. 61–65.

Mosyagin, J. (2010). “Using 4G wireless technology in the car”. In: *Transparent Optical Networks (ICTON), 2010 12th International Conference on*. IEEE, pp. 1–4.

Nebro, A., J. Durillo, J. Garcia-Nieto, C. Coello, F. Luna, and E. Alba (2009). “SMPSO: A new PSO-based metaheuristic for multi-objective optimization”. In: *IEEE Symposium on Computational intelligence in miulti-criteria decision-making, 2009*. Pp. 66–73.

NEO (2015). *VANET/ITS Website (NEO)*. [online] Available in http://neo.lcc.uma.es/staff/jamal/vanet/. Retrieved October 2015.

Nguyen, D. and P. Minet (2007). “Analysis of MPR Selection in the OLSR Protocol”. In: *Advanced Information Networking and Applications Workshops, International Conference on* 2, pp. 887–892.

NS2 (2015). *The Network Simulator NS-2*. [online] Available in http://www.isi.edu/nsnam/ns/. Retrieved October 2015.

Nuaymi, L. (2007). *WiMAX: Technology for Broadband Wireless Access*. Wiley Publishing.

Olariu, S. and M. C. Weigle (2009). *Vehicular networks: from theory to practice*. Crc Press.

Osman, I. H. and J. P. Kelly (2012). *Meta-heuristics: theory and applications*. Springer Science & Business Media.

Paier, A., D. Faetani, and C. Mecklenbrauker (2010). “Performance evaluation of IEEE 802.11p physical layer infrastructure-to-vehicle real-world measurements”. In: *Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on*, pp. 1–5.

Park, Y., J. Ha, S. Kuk, H. Kim, C.-J. Liang, and J. Ko (2014). "A Feasibility Study and Development Framework Design for Realizing Smartphone-Based Vehicular Networking Systems". In: *Mobile Computing, IEEE Transactions on* 13.11, pp. 2431–2444.

Patil, K. V. and M. R. Dhage (2013). "The Enhanced Optimized Routing Protocol for Vehicular Ad hoc Network". In: *International Journal of Advanced Research in Computer and Communication Engineering* 2.10, pp. 4013–4017.

Patil, P. and A. Gokhale (2013). "Voronoi-based placement of road-side units to improve dynamic resource management in VANETs". In: *Int. Conf. on Collaboration Technologies and Systems*, pp. 389–396.

Paton, R. (1994). *Computing with biological metaphors*. Chapman & Hall.

Paul, U., R. Crepaldi, J. Lee, S.-J. Lee, and R. Etkin (2011). "Characterizing WiFi link performance in open outdoor networks". In: *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, pp. 251–259.

Perkins, C., E. Royer, and S. Das (2003). *Ad hoc on demand distance vector (AODV) routing (RFC 3561)*. Tech. rep. [Online]. Available: http://tools.ietf.org/html/rfc3561. IETF MANET Working Group (August. 2003).

Pinart, C., P. Sanz, I. Lequerica, D. García, I. Barona, and D. Sánchez-Aparisi (2008). "DRIVE: a reconfigurable testbed for advanced vehicular services and communications." In: *TRIDENTCOM*. Ed. by M. P. de Leon. ICST, pp. 1–16.

Price, K., R. M. Storn, and J. A. Lampinen (2005). *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Proxim (2015). *Proxim Wireless*. [online] Available in http://www.proxim.com/. Retrieved October 2015.

QMIC (2015). *Qatar Mobility Innovations Center (QMIC)*. [online] Available in http://www.qmic.com/. Retrieved October 2015.

Rana, H., P. Thulasiraman, and R. Thulasiram (2013). "MAZACORNET: Mobility aware zone based ant colony optimization routing for VANET". In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 2948–2955.

Reis, A. B., S. Sargento, F. Neves, and O. Tonguz (2014). "Deploying roadside units in sparse vehicular networks: what really works and what does not". In: *Vehicular Technology, IEEE Transactions on* 63.6, pp. 2794–2806.

Rezaei, S., R. Sengupta, and H. Krishnan (2007). "Reducing the Communication Required By DSRC-Based Vehicle Safety Systems". In: *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 361–366.

Riley, G. F. and T. R. Henderson (2010). "The ns-3 Network Simulator Modeling and Tools for Network Simulation". In: *Modeling and Tools for Network Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. 2.

roadME (2011). *roadME: Fundamentals for Real World Applications of Metaheuristics: The vehicular case*. [online] Available in http://roadme.lcc.uma.es/. Retrieved October 2015.

Rondinone, M., J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Röckl, L. Lin, et al. (2013). "iTETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications". In: *Simulation Modelling Practice and Theory* 34, pp. 99–125.

Ruiz, P. e. a. (2011). "Optimisation of the enhanced distance based broadcasting protocol for MANETs". In: *The Journal of Supercomputing*, pp. 1–28.

S. Bitam, A. M. (2011). "QoS Swarm Bee Routing Protocol for Vehicular Ad Hoc Networks". In: *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–5.

— (2014). *Bio-Inspired Routing Protocols for Vehicular Ad Hoc Networks*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Said, S. and M. Nakamura (2014). “Master-Slave Asynchronous Evolutionary Hybrid Algorithm and ITS Application in VANETs Routing Optimization”. In: *3rd International Conference on Advanced Applied Informatics (IIAIAAI), 2014*, pp. 960–965.

Saleet, H., R. Langar, O. Basir, and R. Boutaba (2009). “Adaptive Message Routing with QoS Support in Vehicular Ad Hoc Networks”. In: *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6.

Saleet, H., R. Langar, K. Naik, R. Boutaba, A. Nayak, and N. Goel (2011). “Intersection-Based Geographical Routing Protocol for VANETs: A Proposal and Analysis”. In: *Vehicular Technology, IEEE Transactions on* 60.9, pp. 4560–4574.

Samsung (2015). *Samsung Electronics*. [online] Available in http://www.samsung.com/. Retrieved October 2015.

Sanchez, M. I., M. Gramaglia, C. J. Bernardos, A. de la Oliva, and M. Calderon (2014). “On the implementation, deployment and evaluation of a networking protocol for VANETs: The VARON case”. In: *Ad Hoc Networks* 19.0, pp. 9 –27.

Sangeeta, K. and K. Sing (2011). “Energy Efficient Routing In MANET Using OLSR”. In: *International Journal on Computer Science and Engineering* 3.16 (4), pp. 1418–1421.

Santa, J., M. Tsukada, T. Ernst, O. Mehani, and A. F. Gómez-Skarmeta (2009). “Assessment of VANET multi-hop routing over an experimental platform”. In: *Int. J. Internet Protoc. Technol.* 4.3, pp. 158–172.

Santa, J., A. Moragón, and A. F. Gómez-Skarmeta (2008). “Experimental evaluation of a novel vehicular communication paradigm based on cellular networks”. In: *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, pp. 198–203.

Santamaria, A. F., C. Sottile, F. De Rango, and S. Marano (2015). “Safety Enhancement and Carbon Dioxide (CO2) Reduction in VANETs”. In: *Mobile Networks and Applications* 20.2, pp. 220–238.

Sattari, M. R. J., R. M. Noor, and H. Keshavarz (2012). “A taxonomy for congestion control algorithms in Vehicular Ad Hoc Networks”. In: *Communication, Networks and Satellite (ComNetSat), 2012 IEEE International Conference on*. IEEE, pp. 44–49.

Saunders, S. and A. Aragon (1999). *Antennas and Propagation for Wireless Communication Systems*. New York, NY, USA: Wiley.

Schmidt, R., T. Leinmuller, E. Schoch, F. Kargl, and G. Schafer (2010). “Exploration of adaptive beaconing for efficient intervehicle safety communication”. In: *Network, IEEE* 24.1, pp. 14–19.

Segura, C., A. Cervantes, A. J. Nebro, M. D. Jaraíz-Simón, E. Segredo, S. García, F. Luna, J. A. Gómez-Pulido, G. Miranda, C. Luque, E. Alba, M. A. Vega-Rodríguez, C. León, and I. M. Galván (2009). “Optimizing the DFCN Broadcast Protocol with a Parallel Cooperative Strategy of Multi-Objective Evolutionary Algorithms”. In: *Evolutionary Multi-Criterion Optimization*. Ed. by M. Ehrgott, C. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux. Vol. 5467. LNCS. Springer Berlin Heidelberg, pp. 305–319.

Sengupta, R., S. Rezaei, S. E. Shladover, D. Cody, S. Dickey, and H. Krishnan (2007). “Cooperative collision warning systems: concept definition and experimental implementation”. In: *Journal of Intelligent Transportation Systems* 11.3, pp. 143–155.

Sethi, A. S. and V. Y. Hnatyshin (2012). *The Practical OPNET User Guide for Computer Network Simulation*. Chapman & Hall/CRC.

Sheskin, D. J. (2007). *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC.

Shrivastava, P., S. Ashai, A. Jaroli, and S. Gohil (2012). “Vehicle-to-Road-Side-Unit Communication Using Wimax”. In: *International Journal of Engineering Research and Applications (IJERA)* 2.4, pp. 1653–1655.

Silva, C. M., A. L. Aquino, and W. M. Jr. (2015). "Deployment of roadside units based on partial mobility information". In: *Computer Communications* 60, pp. 28 –39.

Sommer, C., R. German, and F. Dressler (2011). "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis". In: *Mobile Computing, IEEE Transactions on* 10.1, pp. 3–15.

Stolfi, D. H. and E. Alba (2013). "Red Swarm: Smart Mobility in Cities with EAS". In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. GECCO '13. Amsterdam, The Netherlands: ACM, pp. 1373–1380.

Taherkhani, N. and S. Pierre (2012). "Congestion Control in Vehicular Ad Hoc Networks Using Meta-heuristic Techniques". In: *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*. DIVANet '12. Paphos, Cyprus: ACM, pp. 47–54.

— (2015). "Improving dynamic and distributed congestion control in vehicular ad hoc networks". In: *Ad Hoc Networks* 33, pp. 112–125.

Tan, G. and A.-M. Kermarrec (2012). "Greedy Geographic Routing in Large-scale Sensor Networks: A Minimum Network Decomposition Approach". In: *Networking, IEEE/ACM Transactions on* 20.3, pp. 864–877.

Tanenbaum, A. (2002). *Computer Networks*. 4th. Prentice Hall Professional Technical Reference.

Tielert, T., D. Jiang, H. Hartenstein, and L. Delgrossi (2013). "Joint power/rate congestion control optimizing packet reception in vehicle safety communications". In: *Proceeding of the tenth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM, pp. 51–60.

TomTom (2003). *TomTom*. [online] Available in http://www.qmic.com/. Retrieved October 2015.

Torrent-Moreno, M., J. Mittag, P. Santi, and H. Hartenstein (2009). "Vehicle-to-vehicle communication: fair transmit power control for safety-critical information". In: *Vehicular Technology, IEEE Transactions on* 58.7, pp. 3684–3703.

Toutouh, J. and E. Alba (2011a). "An efficient routing protocol for green communications in vehicular ad-hoc networks". In: *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. GECCO '11. Dublin, Ireland: ACM, pp. 719–726.

— (2011b). "Optimizing OLSR in VANETs with Differential Evolution: A Comprehensive Study". In: *First ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet '11)*. Florida, USA: ACM, pp. 1–8.

— (2011c). "Performance analysis of optimized VANET protocols in real world tests". In: *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference*. IWCMC. Istanbul, Turkey: IEEE, pp. 1244–1249.

— (2012a). "Green OLSR in VANETs with differential evolution". In: *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. ACM, pp. 11–18.

Toutouh, J. and E. Alba (2012b). "Multi-objective OLSR optimization for VANETs". In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*. Barcelona, Spain, pp. 571–578.

— (2012c). "Parallel Swarm Intelligence for VANETs Optimization". In: *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012 Seventh International Conference on*. IEEE, pp. 285–290.

— (2013a). "Computación Natural en Redes Vehiculares". In: *Doctoral Consortium de Multiconferencia CAEPIA - XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA'13)*. Madrid (Spain), pp. 1740–1745.

— (2013b). "Optimizing Telecommunications in Vehicular Networks with a Parallel Multiobjective PSO". In: *22nd International Conference on Multiple Criteria Decision Making (MCDM2013)*, p. 295.

— (2015a). “Comunicación efficiente entre vehículos aplicando un algoritmo multi-objetivo paralelo”. In: *Actas del VII Congreso sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'2015)*. Mérida (Spain), pp. 503–510.
— (2015b). “Metaheuristics for energy-efficient data routing in vehicular networks”. In: *International Journal of Metaheuristics* 4.1, pp. 27–56.
— (2015c). “Parallel multi-objective metaheuristics for smart communications in vehicular networks”. English. In: *Soft Computing*. In Press., pp. 1–13. URL: http://dx.doi.org/10.1007/s00500-015-1891-2.
— (2016). “Light commodity devices for building vehicular ad hoc networks: An experimental study”. In: *Ad Hoc Networks* 37, Part 2, pp. 499–511.
Toutouh, J., J. García-Nieto, and E. Alba (2010a). “Configuración Óptima del Protocolo de Encaminamiento OLSR para VANETs Mediante Evolución Diferencial”. In: *Actas del VII Congreso sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'10)*. Valencia (Spain), pp. 463–471.
— (2010b). “Optimal configuration of OLSR routing protocol for VANETs by means of Differential Evolution”. In: *3rd International Conference on Metaheuristics and Nature Inspired Computing, META 2010*. D'Jerba (Tunissia), pp. 1–2.
Toutouh, J., S. Nesmachnow, and E. Alba (2012a). “Evolutionary Power-Aware Routing in VANETs using Monte-Carlo Simulation”. In: *Proceedings of The 10th International Conference on High Performance Computing and Simulation (HPCS 2012)*. Madrid, Spain: IEEE Computer Society Press, pp. 119–125.
Toutouh, J., J. García-Nieto, and E. Alba (2012b). “Intelligent OLSR Routing Protocol Optimization for VANETs”. In: *Vehicular Technology, IEEE Transactions on* 61.4, pp. 1884–1894.
Toutouh, J., S. Nesmachnow, and E. Alba (2013). “Fast energy-aware OLSR routing in VANETs by means of a parallel evolutionary algorithm”. In: *Cluster Computing* 16.3, pp. 435–450.
Trullols, O., M. Fiore, C. Casetti, C. Chiasserini, and J. Ordinas (2010). “Planning roadside infrastructure for information dissemination in intelligent transportation systems”. In: *Computer Communications* 33.4, pp. 432–442.
Unex (2015). *DCMA-86P2 Network Interface Card*. [online] Available in http://www.unex.com.tw/wifi-module-fcc/dcma-86p2. Retrieved October 2015.
Uzcategui, R and G. Acosta-Marum (2009). “WAVE: a tutorial”. In: *Communications Magazine, IEEE* 47.5, pp. 126–133.
Van Arem, B., C. J. Van Driel, and R. Visser (2006). “The impact of cooperative adaptive cruise control on traffic-flow characteristics”. In: *Intelligent Transportation Systems, IEEE Transactions on* 7.4, pp. 429–436.
Vega, M., J. Gómez, E. Alba, D. Vega, S. Priem, and G. Molina (2007). “Evaluation of Different Metaheuristics Solving the RND Problem”. English. In: *Applications of Evolutionary Computing*. Vol. 4448. Lecture Notes in Computer Science. Springer, pp. 101–110.
Veldhuizen, D. A. V. and G. B. Lamont (1998). *Multiobjective evolutionary algorithm research: A history and analysis*. Tech. rep. TR-98-03. Wright-Patterson, USA: Graduate School of Engineering, Air Force Institute of Technology.
Villalba, L. G., D. R. Cañas, and A. L. S. Orozco (2010). “Bio-inspired routing protocol for mobile ad hoc networks”. In: *IET communications* 4.18, pp. 2187–2195.
Vinel, A. (2012). “3GPP LTE versus IEEE 802.11 p/WAVE: which technology is able to support cooperative vehicular safety applications?” In: *Wireless Communications Letters, IEEE* 1.2, pp. 125–128.
VSC (2005). *Vehicle Safety Communications Project*. Tech. rep. [Online]. Available: http://www.its.dot.gov/research_docs/pdf/59vehicle-safety.pdf. Vehicle Safety Communications Consortium.

Wang, C., X. Li, F. Li, and H. Lu (2014). "A mobility clustering-based roadside units deployment for VANET". In: *16^{th} Asia-Pacific Network Operations and Management Symposium*, pp. 1–6.

Wang, J., E. Osagie, P. Thulasiraman, and R. K. Thulasiram (2009). "HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network ". In: *Ad Hoc Networks* 7.4. I. Bio-Inspired Computing and Communication in Wireless Ad Hoc and Sensor Networks II. Underwater Networks, pp. 690 –705.

Waze (2009). *Waze*. [online] Available in http://www.waze.com/. Retrieved October 2015.

White, D. (2012). "Software review: the ECJ toolkit". English. In: *Genetic Programming and Evolvable Machines* 13.1, pp. 65–67.

Whitehouse, K., C. Karlof, and D. Culler (2007). "A practical evaluation of radio signal strength for ranging-based localization". In: *ACM SIGMOBILE Mobile Computing and Communications Review* 11.1, pp. 41–52.

WHO (2015). *Road Traffic Injuries. World Health Organization Fact Sheet. 2015*. [online] Available in http://www.who.int/mediacentre/factsheets/fs358/en/. Retrieved July 15, 2015.

Wischhof, L. and H. Rohling (2005). "Congestion control in vehicular ad hoc networks". In: *Vehicular Electronics and Safety, 2005. IEEE International Conference on*, pp. 58–63.

Wisitpongphan, N., O. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar (2007). "Broadcast storm mitigation techniques in vehicular ad hoc networks". In: *Wireless Communications, IEEE* 14.6, pp. 84–94.

Wu, S.-H., C.-M. Chen, and M.-S. Chen (2010). "An Asymmetric and Asynchronous Energy Conservation Protocol for Vehicular Networks". In: *Mobile Computing, IEEE Transactions on* 9.1, pp. 98–111.

Xu, H. and M. Barth (2004). "A transmission-interval and power-level modulation methodology for optimizing inter-vehicle communications". In: *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. ACM, pp. 97–98.

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

Yang, X.-S., S. F. Chien, and T. O. Ting (2015). *Bio-inspired Computation in Telecommunications*. Morgan Kaufmann.

Yousefi, S., M. Mousavi, and M. Fathy (2006). "Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives". In: *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pp. 761–766.

Zavala, G. R., A. J. Nebro, J. J. Durillo, and F. Luna (2014). "Integrating a multi-objective optimization framework into a structural design software". In: *Advances in Engineering Software* 76, pp. 161–170.

Zeadally, S., R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan (2012). "Vehicular ad hoc networks (VANETS): status, results, and challenges". English. In: *Telecommunication Systems* 50.4, pp. 217–241.

Zhao, J. and R. Govindan (2003). "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks". In: *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. SenSys '03. Los Angeles, California, USA: ACM, pp. 1–13.

Zitzler, E. and L. Thiele (1999). "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach". In: *Evolutionary Computation, IEEE transactions on* 3.4, pp. 257–271.

Zukarnain, Z., N. Al-Kharasani, S. Subramaniam, and Z. Hanapi (2014). "Optimal Configuration for Urban VANETs Routing using Particle Swarm Optimization". In: *Proceeding of the International Conference on Artificial Intelligence and Computer Science 2014*, pp. 1–6.

Ancient brooch used in the Berber/Amazigh culture of Morocco.