

Infrastructure Deployment in Vehicular Communication Networks Using a Parallel Multiobjective Evolutionary Algorithm

Renzo Massobrio^{1*}, Jamal Toutouh², Sergio Nesmachnow¹, Enrique Alba²

¹Universidad de la República, Herrera y Reissig 565, Montevideo, 11300, Uruguay

²Dept. de Lenguajes y Ciencias de la Computación, University of Málaga, Málaga, 29071, Spain

This article describes the application of a multiobjective evolutionary algorithm for locating roadside infrastructure for vehicular communication networks over realistic urban areas. A multiobjective formulation of the problem is introduced, considering quality-of-service and cost objectives. The experimental analysis is performed over a real map of Málaga, using real traffic information and antennas, and scenarios that model different combinations of traffic patterns and applications (text/audio/video) in the communications. The proposed multiobjective evolutionary algorithm computes accurate trade-off solutions, significantly improving over state-of-the-art algorithms previously applied to the problem. © 2016 Wiley periodicals, Inc.

...

KEY WORDS: VANETs, infrastructure placement, multiobjective evolutionary algorithms, smart cities.

1. INTRODUCTION

Vehicular traffic is a major concern in modern cities.¹⁴ Several problems related to mobility, traffic safety, environment, etc. can be efficiently solved by applying smart computational methods. In this context, the concept of *smart cities* has emerged as a key issue in modern urbanization. A smart city applies information technologies to enhance quality, performance, and interactivity of urban services and/or to reduce costs and resource consumption. Road traffic management is a specific area that makes use of smart city applications, with the goal of improving the management of urban flows and allowing for real time responses to challenges that have great impact on the citizens.¹¹

A number of smart city solutions are based on *intelligent transport systems* (ITS). The main idea behind these systems consist in sharing information about the traffic conditions with road users and authorities. A better informed citizen can take better driving decisions, positively influencing the global traffic safety, efficiency.

Vehicular ad hoc networks (VANETs) emerge as a promising technology to allow continuous data exchange between vehicles equipped with an on-board unit (OBU). Vehicles can also communicate with *roadside unit* (RSU) elements via *direct short range communications* (DSRC). Depending on the type of nodes involved, several types of communications can occur in a VANET: vehicle-to-vehicle (V2V) communications, when the vehicles communicate directly with each other, and vehicle-to-infrastructure (V2I) communications, when the vehicles exchange data with RSUs.

VANETs allow developing a large set of powerful applications to improve road transport experience for both drivers and passengers. Typically, these applications are categorized into *safety* and *non-safety* applications. The first ones aim at improving road safety and avoiding hazardous situations (road accidents), e.g., cooperative driving and intersection collision avoidance

* Author to whom all correspondence should be addressed: e-mail: renzom@fing.edu.uy

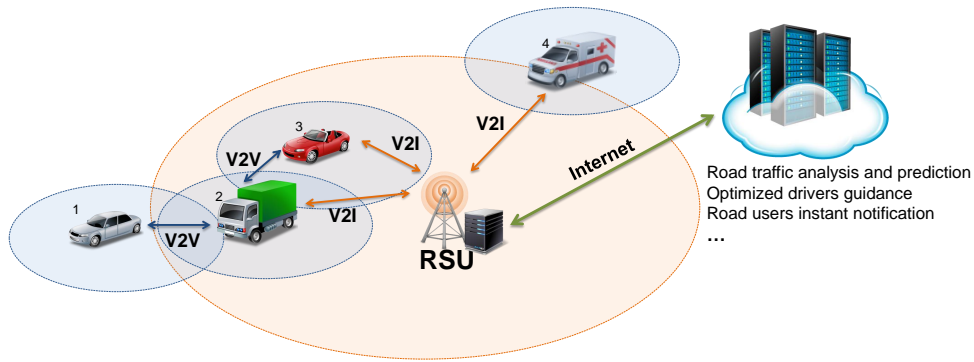


Figure 1. Global VANET architecture.

applications. Non-safety applications include a collection of different solutions oriented to enhancing the traffic efficiency (e.g., travel times, fuel consumption, CO₂ emissions, etc.). These applications also allow improving the comfort and entertainment of passengers.

Safety and traffic efficiency applications, such as *Cooperative Vehicle Safety*, gather real-time data from diverse sources (vehicle sensors, information received from other nodes, or both), process it, and disseminate it to the other nodes. Most of these applications rely on periodic message broadcasting or beaconing. This kind of applications require very short data delivery times (in terms of milliseconds),⁵ since larger delivery times increase the uncertainty on the system and may cause hazardous situations. Infotainment VANET applications (e.g., audio or video broadcasting, on-line gaming, etc.) mainly rely on continuous data streams. The real time requirements are lower than for safety and efficiency applications, but they require larger transmission data rates (in terms of tenths of kilobytes per second) to keep the quality of the service provided.

In this study, we focus on a specific element of VANET architecture, the RSUs, which are devices that are usually installed along the roads on the roadside infrastructure elements, e.g., traffic lights. In addition, they may be fixed along roadside as specific dedicated VANET elements. RSUs 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. RSUs perform three main functions: i) acting as an information transmitter or receiver in VANET applications, e.g., warning about of the existence of roadworks, accidents, etc.; 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 other nodes in the VANET.

Figure 1 illustrates a typical VANET scenario and the importance of including RSUs in the VANET architecture. In the figure, the coverage of the OBUs is shaded in blue, and represents the maximum distance in which two vehicles may utilize V2V to communicate with each other (i.e., vehicles 2 and 3 and vehicles 1 and 2 are the only ones that are able to exchange information with each other). The communication range of the RSU is shaded in orange, and therefore, vehicles 2, 3, and 4 can communicate with the RSU via V2I. In the presented example, the ambulance (vehicle 4) is approaching to vehicles 1, 2, and 3. These vehicles are outside the coverage of the OBU of the ambulance, therefore V2V communications cannot be used. The only way to warn that the ambulance is approaching is by using a RSU to forward messages to vehicles 2 and 3. Thus, the RSU extends the effective communication range of the ambulance. In turn, the RSU can inform vehicles about possible new and more efficient routes, helping to improve the ambulance trip. Furthermore, all vehicles in the scenario can use the RSU connectivity to access traffic services, Internet, etc.

Consequently, the deployment of a fixed infrastructure of RSUs along the roads is of vital importance when deploying modern and powerful ITS, which helps to mitigate the serious road traffic problems that have to be confronted in modern cities.

Deploying the RSU infrastructure for VANETs is a challenge because network designers must decide about the number, type, and location of RSUs to maximize quality-of-service (QoS) of the

VANET, while satisfying and/or minimizing the deployment cost requirements. At this point, the network designers have to take into account that VANETs are used by different types of applications and services (safety and non-safety), and therefore, the final effective QoS of the network has to satisfy the required transmission data rates and delivery times of such applications.

The RSU Deployment Problem (RSU-DP) consists in placing a set of RSU terminals in a given area. We study a multiobjective version of the RSU-DP, which proposes maximizing the network QoS and minimizing the deployment costs. This is a hard-to-solve optimization problem on city-scaled areas, as the number of possible solutions is very large.³¹ Heuristics and metaheuristics²⁸ are promising methods to deal with the RSU-DP; they allow computing good infrastructure designs in reduced execution times.^{6,34} In this article, we propose applying the NSGA-II evolutionary algorithm¹³ to design the RSU infrastructure within a city-scaled network in Málaga (Spain). In order to obtain realistic results, we consider real information about road traffic (road map and traffic flow), hardware (network capabilities and costs), and VANET applications.

This article extends our previous conference paper,²⁴ where the problem was first presented and preliminary results of applying a multiobjective evolutionary algorithm (MOEA) were reported.

The main contributions of the research reported in this article are: i) the multiobjective formulation of the RSU-DP considers in the QoS evaluation the maximum number of vehicles that can be simultaneously attended by a given RSU type, extending our previous study that just took into account the effective radio range of each RSU type; ii) we solve realistic scenarios, larger than those previously solved in the related literature, and we include here the last new and updated traffic data published by the Málaga city council for 2015; iii) we model a set of real VANET applications, considered in the QoS metric applied in the problem formulation to evaluate a set of potential locations for RSUs (these applications were not present in the previous conference paper); iv) we adapt two state-of-the-art heuristic methods (deterministic and randomized) for the problem, to be used as a baseline for the comparison of the proposed MOEA; v) we propose a parallel master-slave MOEA, including a new initialization operator based on the Randomized Knapsack algorithm as a novelty regarding our previous study; and vi) we report accurate results for cost and QoS for the problem instances solved: the proposed NSGA-II is able to improve over the results computed by the best baseline heuristics up to 24.68% and 52.71% in terms of cost, and up to 34.09% and 39.48% in terms of QoS.

The article is organized as follows. Section 2 introduces the multiobjective version of the RSU-DP. Section 3 presents a review of works solving the RSU location problem and related radio network design problems. Section 4 introduces the methods applied to solve the problem. The specific features of the proposed MOEA to solve the RSU-DP are described in Section 5. Section 6 describes the heuristic methods proposed as a baseline for comparing the results computed using the proposed MOEA, and reports the experimental evaluation of the proposed method on a set of realistic scenarios in the city of Málaga, using real infrastructure and VANET applications. Finally, Section 7 formulates the conclusions and the main lines for future work.

2. THE RSU DEPLOYMENT PROBLEM

The mathematical formulation of the RSU-DP considers the following elements:

- A set of RSUs $R = \{R_1, \dots, R_q\}$ to be installed in a city scenario for providing efficient VANET communications.
- A set of RSU types $T = \{t_1, t_2, \dots, t_l\}$. 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 $type: R \rightarrow T$.
- A set of road segments $S = \{s_1, s_2, \dots, s_n\}$, which are potential locations for placing RSUs along the city streets. Each segment s_i is defined by a pair of points (p_j, p_k) , with $p_j, p_k \in P = \{p_1, p_2, \dots, p_m\}$. Each point p_j is identified by its geographical coordinates (latitude, longitude). The length of a given segment s_i is given by the function $len: S \rightarrow \mathbb{R}^+$. RSUs can be placed at any location within each segment s_i .

- An estimation of the number of vehicles per time period across each segment s_i , given by function $NV: S \rightarrow \mathbb{N}^+$, and the average vehicle speed for each segment, given by function $sp: S \rightarrow \mathbb{R}^+$.
- A cost function $C: T \rightarrow \mathbb{R}^+$, where $C(t_g)$ indicates the monetary cost of placing a RSU of type t_g in the deployed infrastructure.
- A set of applications $A = \{A_1, A_2, \dots, A_u\}$ to be used over the VANET. Each application has specific QoS requirements, given by function $Q: A \rightarrow \mathbb{N}^+ \times \mathbb{N}^+$. $Q(A_h)$ is a vector with two elements, indicating the QoS requirements for *packet delivery ratio* (PDR) and *end-to-end delay* (E2ED) for application A_h . On a given scenario, $Q(A_h)$ is used to define the *maximum number of users* to be served by each RSU, given by function $MU: R \times A \rightarrow \mathbb{N}^+$.

Solutions of the problem are defined by a set of RSUs placed over the road segments of the city, represented by a set $sol = \{R_1, R_2, \dots, R_l\}$, where l is the number of RSUs (*#RSU*) in solution sol ($l \leq n$). Each RSU is installed in a specific coordinate within a segment s_i . The segments covered by a RSU are given by the function $cov: R \rightarrow S$, and the portion of segment s_k covered by RSU R_j is given by the function $cp: R \times S \rightarrow [0, 1]$.

The multiobjective version of the RSU-DP 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, while simultaneously minimizing the *total cost* of deployment. The service time is a metric related to the QoS offered to the VANET users. It is related to the number of vehicles attended by RSUs, the time they are served (considering the coverage and average speed per each road segment), and the type of applications used in the studied scenario.

Formally, the problem is defined as the simultaneous optimization of two objective functions: maximize the QoS, given by $f_1(sol, A_h)$ (Equation 1) and minimize the cost, given by $f_2(sol)$ (Equation 2). The corresponding values for function MU for each RSU and application type are computed by simulations (see Section 6.2).

$$\max f_1(sol, A_h) = \sum_{R_j \in sol} \max \left(MU(R_j, A_h), \sum_{s_i \in cov(R_j)} NV(s_i) \times \frac{cp(R_j, s_i) \times len(s_i)}{sp(s_i)} \right) \quad (1)$$

$$\min f_2(sol) = \sum_{R_j \in sol} C(type(R_j)) \quad (2)$$

3. RELATED WORK

Including RSUs in the network loop improves the global VANET performance in terms of connectivity, transmission delays, and communication ranges.²⁰ Deploying a low cost and high coverage RSU infrastructure is often a capital issue for the success of VANETs in real cities. This section reviews computational intelligence methods applied to the RSU-DP and related problems.

In the related literature, different studies address the RSU-DP. Most of these works analyze RSU-DP as a version of the Radio Network Design problem.²⁵ 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.

Both exact methods and heuristics have been applied to solve the RSU-DP and related problems.

Trullols et al.³³ defined the Maximum Coverage with Time Threshold Problem (MCTTP) to maximize the number of vehicles that get in contact with a given number of RSUs for a given amount of time over a certain area. The authors proposed three greedy algorithms with different knowledge of the road topology and identity of the vehicles. These approaches were applied over a scenario with real road and mobility data from Zurich, Switzerland. The results showed that knowledge of vehicular mobility is the main factor to achieve an almost-optimal roadside deployment. Given such knowledge, the heuristics successfully planned a deployment capable of informing more than 95% of vehicles.

Aslam et al.² applied the Balloon Expansion Heuristic (BEH) and Binary Integer Programming (BIP) to minimize the reporting time, installing a fixed number of RSUs in Miami, USA. The methods used information about speed, traffic density, and likelihood of incidents. BEH performed better than BIP in the reported experiments.

A Voronoi-based algorithm was applied by Patil and Gokhale³⁰ to optimize packet loss, communications delays, and network coverage, while minimizing the number of RSUs required in a deployed vehicular network in an area of Nashville, USA. The algorithm used information about the speed of vehicles and the traffic density to evaluate the solutions.

Ben Brahim et al.⁴ solved a variant of the RSU-DP in Doha, Qatar, considering the traffic network as a graph with weighted links. The weight of the links are computed according to road traffic and mobility-based parameters, such as road traffic density and average speed. Afterwards, all potential positions for the RSUs are computed by applying two different approaches: a dynamic algorithm based on 0-1 Knapsack problem (KP_DynAlg) solver and the PageRank algorithm. The KP_DynAlg improved over the results computed by the PageRank method.

Some studies have proposed applying evolutionary algorithms (EAs) for solving variants of the RSU-DP, in order to obtain accurate solutions while consuming reasonable computational resources. An early approach studied applying a genetic algorithm (GA) that uses a VANET simulator to evaluate the QoS of the computed solutions in a given area of $16 \times 16 \text{ km}^2$ in the city of Brunswick, Germany, with about 500 km of roads and 10000 vehicles.²¹ The authors introduce a domain aggregation scheme to minimize the required overall bandwidth for the VANET and propose a GA to locate static roadside units (called *supporting units*) to deal with a highly partitioned VANET in an early deployment stage. The proposed GA was useful to improve the travel time savings achieved by a given vector of active SU locations.

Cavalcante et al.⁶ compared GA against the greedy approach proposed by Trullols et al.³³ to solve the MCTTP, taking into account real data from four different regions: Zurich downtown, Winterthur, Baden, and Baar. The proposed GA uses a greedy method to initialize the population. The results showed that the GA solutions obtained better vehicle coverage: up to 11% better than those computed by the greedy approach.

Another GA proposal is by Cheng et al.,⁹ who used geometry-based coverage information about the roads (without vehicles mobility related data) of Yukon Territory, Canada, for the solution evaluation. The GA computed the fitness in terms of the ratio between the covered road area and the whole road area, computed using a square grid of $1m \times 1m$. This approach improved the results computed by the α -coverage algorithm, which proposes placing the RSUs in the center of the junctions.

A summary of the main related works about heuristics and computational intelligence methods (following evolutionary and non-evolutionary approaches) to solve the RSU-DP and related problems is presented in Table I.

Table I. Summary: related work about heuristics and computational intelligence methods applied to the RSU-DP.

<i>author</i>	<i>year</i>	<i>problem</i>	<i>method</i>	<i>scenario</i>
non-evolutionary approaches				
Trullols et al. ³³	2010	vehicle maximization	greedy algorithms	Zurich, Switzerland
Aslam et al. ²	2012	RSU installation	BEH, BIP	Miami, USA
Patil and Gokhale ³⁰	2013	RSU optimization	Voronoi-based algorithm	Nashville, USA
Ben Brahim et al. ⁴	2014	mobility/traffic based	Knapsack, PageRank	Doha, Qatar
evolutionary approaches				
Lochert et al. ²¹	2008	supporting units location	GA	Brunswick, Germany
Cavalcante et al. ⁶	2012	MCTTP-RSU-DP	GA	Switzerland cities
Cheng et al. ⁹	2013	geometry-based coverage	GA	Yukon territory
Massobrio et al. ^{23,24}	2015	multiobjective RSU-DP	MOEA	Málaga, Spain

Our previous works^{23,24} were the first studies that applied an explicit multiobjective approach to solve the RSU-DP. Our proposal was oriented to maximize the coverage, in terms of the time that vehicles are connected to the RSUs, and minimize the deployment cost. We consider real information concerning both traffic (speed, traffic density, and road map) and hardware (costs and capabilities) for the case of urban locations in Málaga, Spain. The proposed MOEA obtained significantly better results than ad hoc greedy approaches, but the computed solutions did not cover the map properly, focusing on streets with high number of vehicles instead.

In this article, we extend our previous work²⁴ by considering a more realistic QoS model that includes a set of different VANET applications, which are taken into account to compute the maximum number of vehicles that can be simultaneously attended by a given RSU type. A more comprehensive experimental analysis is performed, including updated traffic data and modeling real VANET applications. Finally, the results are compared against those computed using specific heuristics, adapted from the work of Ben Brahim et al.,⁴ in terms of cost, QoS, and multiobjective optimization metrics.

4. METAHEURISTICS AND EVOLUTIONARY COMPUTATION

This section introduces the methods applied to solve the problem: metaheuristics, evolutionary algorithms and multiobjective evolutionary algorithms.

4.1. Metaheuristics

Metaheuristics are strategies to define algorithmic frameworks that allow designing efficient techniques to find approximate solutions for search, optimization, and learning problems.¹⁵ They define high-level, heuristic-based, soft computing methods that can be applied to solve different optimization problems, by instantiating a generic resolution procedure.²⁸

In practice, many optimization problems arising in nowadays real-world applications from science and technology are NP-hard and intrinsically complex. A lot of computing effort is demanded to solve them, due to a number of reasons: they have very large-dimension search spaces, they include hard constraints that make the search space very sparse, they are multimodal or multiobjective problems taking into account hard-to-evaluate optimization functions, or they manage very large volumes of data. This is the case for the problem solved in this article: the deployment of roadside infrastructure for VANETs, which is a variant of the well-known Radio Network Design problem.²⁵

Metaheuristics provide efficient and accurate methods for solving realistic instances of the problem, that often cannot be solved using exact optimization methods that are extremely time-consuming. In this article, we apply a multiobjective evolutionary metaheuristic to solve the RSU-DP. The main features of EAs and their multiobjective variants are described next.

4.2. Evolutionary algorithms

EAs are non-deterministic methods that emulate the evolutionary process of species in nature to solve optimization, search, and other related problems.^{3,16} In the last thirty years, EAs have been successfully applied for solving problems underlying many real and complex applications.

Algorithm 1 shows the generic schema of an EA. It is an iterative technique (each iteration is called a *generation*) that works by applying stochastic operators on a set of *individuals* (the population P) in order to improve their *fitness*, a measure that evaluates how good is a solution to solve the problem. Every individual in the population encodes a candidate solution for the problem. The initial population is generated by a random method or by using a specific heuristic for the problem (line 2 in Algorithm 1). An evaluation function associates a fitness value to every individual (line 4). The search is guided by a probabilistic selection-of-the-best technique (for both parents and offspring) to tentative solutions of higher quality (line 5). Iteratively, solutions are modified by the probabilistic application of *variation operators* (line 6), including the *recombination* of parts from two individuals or random changes (*mutations*) in their contents, which are applied for building new solutions during the search.

The stopping criterion usually involves a fixed number of generations or execution time, a quality threshold on the best fitness value, or the detection of a stagnation situation. Specific policies are used to select the groups of individuals to recombine (the *selection* method) and to determine which new individuals are inserted in the population in each new generation (the *replacement* criterion). The EA returns the best solution ever found in the iterative process, taking into account the fitness function.

Algorithm 1 Generic schema for an EA.

```

1:  $t \leftarrow 0$  ▷ Generation counter
2: initialize( $P(0)$ )
3: while not stopcriterion do
4:   evaluate( $P(t)$ )
5:   parents  $\leftarrow$  selection( $P(t)$ )
6:   offspring  $\leftarrow$  variation operators(parents)
7:    $P(t+1) \leftarrow$  replacement(offspring,  $P(t)$ )
8:    $t \leftarrow t + 1$ 
9: end while
10: return best solution ever found

```

One of the most popular variants of EA in the literature is the genetic algorithm (GA), which has been extensively used to solve optimization problems mainly due to its simplicity and versatility.

The classic GA formulation was presented by Goldberg.¹⁶ Based on the generic schema of an EA, a GA defines selection, recombination and mutation operators, applying them to the population of potential solutions in each generation. In a classic application of a GA, the recombination operator is mainly used to guide the search (by exploiting the characteristics of suitable individuals), while the mutation is used as the operator aimed at providing diversity for exploring different zones of the search space.

Parallel models for metaheuristics and EAs have been proposed to speed up the computing time required for the search when dealing with complex objective functions and hard search spaces.¹ In this work, we apply a master-slave model for parallelization, in order to reduce the execution time of computing the QoS objective when solving the RSU-DP problem (see details in Section 5).

4.3. Multiobjective evolutionary algorithms and NSGA-II

Multiobjective evolutionary algorithms (MOEAs)^{10,13} are specific evolutionary optimization methods conceived to solve problems with many conflicting objective functions. MOEAs have obtained accurate results when used to solve difficult real-life optimization problems in many research areas.

Unlike many traditional methods for multiobjective optimization, MOEAs find a set with several solutions in a single execution, since they work with a population of tentative solutions. MOEAs are designed to fulfill two goals at the same time: *i*) approximate the Pareto front, and *ii*) maintain diversity, instead of converging to a section of the Pareto front. A Pareto-based evolutionary search leads to the first goal, while the second is accomplished using specific techniques from multi-modal function optimization (e.g., sharing, crowding, etc.).

In this work, we apply NSGA-II (Non-dominated Sorting Genetic Algorithm, version II),¹² a popular state-of-the-art MOEA that has been successfully applied in many areas. A schema of NSGA-II is presented in Algorithm 2 (where N is the population size). The fitness calculation is based on Pareto dominance, building *fronts* of solutions. The evolutionary search on NSGA-II improves over the previous version (NSGA), using: *i*) a non-dominated, elitist sorting that reduces the complexity of the dominance check; *ii*) a crowding technique for diversity preservation; and *iii*) a fitness assignment that considers crowding distance values.

The NSGA-II algorithm proposed in this work has been engineered to compute accurate solutions for the RSU-DP. The main implementation details are presented in the next section.

Algorithm 2 Schema of the NSGA-II algorithm.

```

1:  $t \leftarrow 0$  ▷ Generation counter
2: offspring  $\leftarrow \emptyset$ 
3: initialize( $P(0)$ )
4: while not stopcriterion do
5:   evaluate( $P(t)$ )
6:    $R \leftarrow P(t) \cup \text{offspring}$ 
7:   fronts  $\leftarrow$  non-dominated sorting( $R$ )
8:    $P(t+1) \leftarrow \emptyset$ 
9:    $i \leftarrow 1$ 
10:  while  $|P(t+1)| + |\text{fronts}(i)| \leq N$  do
11:    crowding distance (fronts( $i$ ))
12:     $P(t+1) \leftarrow P(t+1) \cup \text{fronts}(i)$ 
13:     $i \leftarrow i+1$ 
14:  end while
15:  sorting by distance (fronts( $i$ ))
16:   $P(t+1) \leftarrow P(t+1) \cup \text{fronts}(i)[1:(N - |P(t+1)|)]$ 
17:  selected  $\leftarrow$  selection( $P(t+1)$ )
18:  offspring  $\leftarrow$  variation operators(selected)
19:   $t \leftarrow t + 1$ 
20: end while
21: return computed Pareto front

```

5. THE PROPOSED NSGA-II ALGORITHM FOR THE RSU-DP

This section presents the details of the proposed NSGA-II evolutionary algorithm for the RSU-DP.

5.1. Solution encoding

In the proposed NSGA-II, solutions are represented as vectors of real numbers, having length $n = \#S$ (the number of elements in the set of road segments S). Each position on the vector contains the information about the RSU to install (if any) on the corresponding segment: i) the type of the RSU is given by the integer part of the real number (0 stands for the absence of RSU in the considered segment, and integers $1 \dots k$ represent types $t_1 \dots t_k$, respectively); and ii) the candidate location to install the RSU 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 2 shows an example encoding for a scenario consisting of four segments, where three RSUs are placed. For instance, the value 1.50 in position 2 of the vector indicate that the solution proposes to install a RSU of type 1 (integer part of 1.50) at the middle (fractional part of 1.50 = 0.50) of segment $s_2 = (p_2, p_3)$. The same holds for value 2.16 in the first position of the vector, corresponding to segment s_1 , where a RSU of type 2 is installed at $0.16 \times \text{len}(s_1)$ within segment $s_1 = (p_1, p_2)$. Finally, the value 0.33 in the fourth position of the vector indicates that the solution proposes not installing a RSU in segment $s_4 = (p_1, p_4)$ (the fractional part of the value encoded is irrelevant in this case). In fact, considering the covering radios of the RSUs placed in the map in Figure 2, we see that it could be a wise decision, because segment s_4 is fully covered by RSUs t_2 and t_3 (of course, this decision reduces the installation cost, but the QoS for users might be reduced, depending on the number of vehicles and the applications used in the considered scenario).

5.2. Evolutionary operators

The proposed MOEA applies different evolutionary operators and a parallel model to efficiently address the RSU-DP problem. This section describes such operators and parallel model.

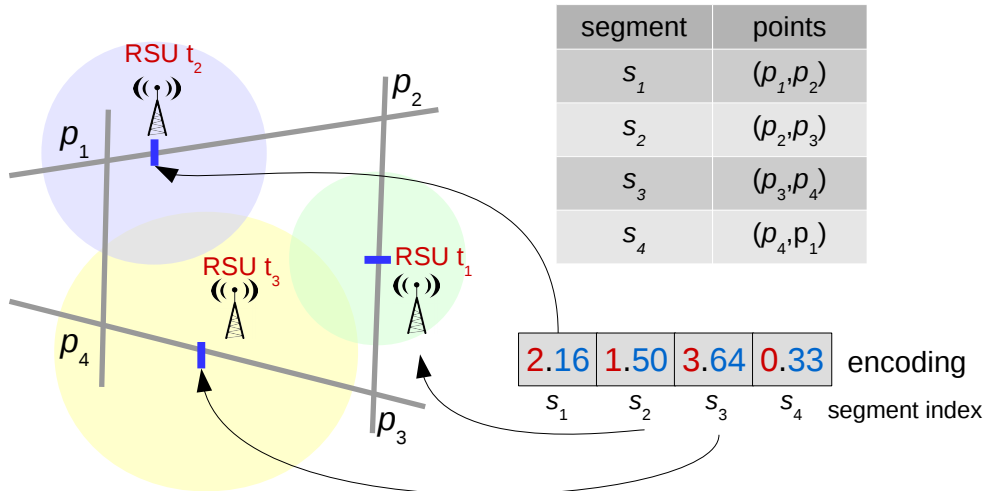


Figure 2. Encoding for RSU-DP solutions.

5.2.1. *Initialization* Instead of starting from a random set of solutions, we decided to seed the initial population with the solutions computed by the Randomized Knapsack heuristic, explained in Section 6.3. This decision allows focusing the evolutionary search on a subspace of good quality solutions. The Randomized Knapsack is a constructive method, providing a range of partial solutions that are useful to generate the initial population of the MOEA. Particularly, we employ solutions with a high number of RSUs, since we are not interested in exploring the areas of the Pareto front with negligible values for QoS (this region has configurations that are not useful in practice, in a real-world scenario).

5.2.2. *Selection* The selection operator used is the tournament selection, as originally proposed in the NSGA-II algorithm.¹³ The tournament size is two individuals and the fittest individual survives.

5.2.3. *Exploitation: recombination* The recombination operator used is the well-known two-point crossover (2PX), where offspring are generated by swapping genes in the parents' chromosomes that fall between two randomly selected cutting points.

5.2.4. *Exploration: mutation* We designed an ad-hoc mutation operator in order to provide enough diversity to the search, avoiding NSGA-II to get stuck in a specific region of the Pareto front. The mutation operator probabilistically applies three variations on solutions. This variations work as follows:

1. With probability π_A , the mutation operator changes the integer part of the selected gene value to 0, thus removing the RSU (if any) from the corresponding segment (see Figure 3a).
2. With probability π_B , the mutation operator changes the integer part of the selected gene value for a different one randomly picked in $[1, k]$, thus changing the type of the RSU (or adding one if there was none) to a random type picked uniformly in T (see Figure 3b).
3. With probability $1 - \pi_A - \pi_B$, the variation applied corresponds to a Gaussian Mutation with a standard deviation of σ to the selected gene value, thus changing the position of the RSU within the segment (see Figure 3c).

5.2.5. *Parallel model* We apply a master-slave parallel model for metaheuristics¹ to reduce the execution time demanded to evaluate the objective functions for each individual in the population. The decomposition approach allows NSGA-II to efficiently compute the objective functions for the set of candidate solutions in the population, as described in the next subsection.

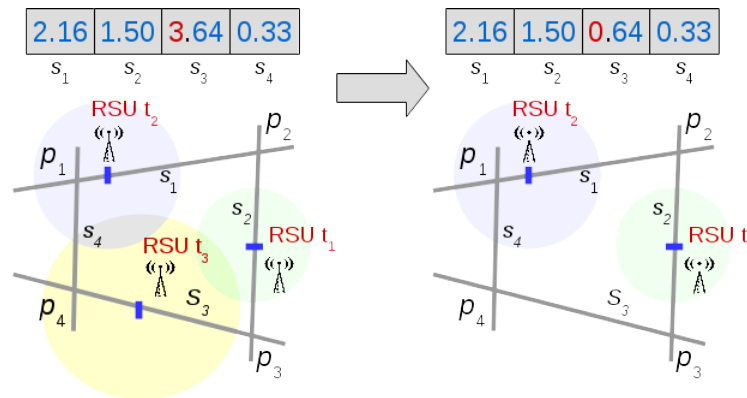
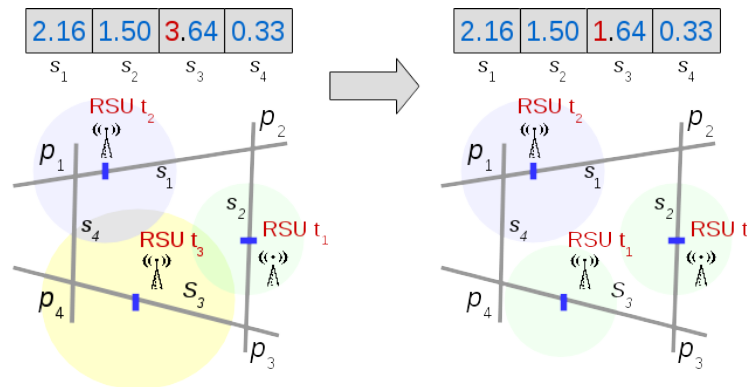
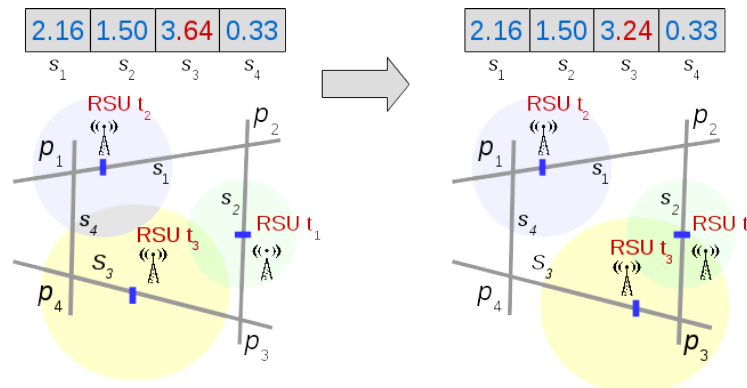
(a) Mutation applied with probability π_A (remove RSU).(b) Mutation applied with probability π_B (modify RSU type).(c) Mutation applied with probability π_C (modify RSU location within segment).

Figure 3. Variations applied by the mutation operator.

5.3. Computing the objective functions

In order to compute the two objective functions to be optimized in RSU-DP, two functions were evaluated: the installation cost and the quality of service. These two functions are presented next.

5.3.1. Installation cost The total installation cost is simply computed by adding the cost of each RSU placed in the solution, taking into account the corresponding RSU type.

5.3.2. *Quality of service* For computing the QoS, we consider the distances and values shown in the diagram in Figure 4 (intersection of streets A and B). The RSU placed in the point “×” 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), both in street B. The number of effective vehicles attended is computed by $\sum_{i=1}^{i=4} NV(s_i) \times \frac{c_i}{sp(s_i)}$. The computation requires finding the intersections between the road segments and the circle defining the coverage of the RSU. Given that the distances involved are relatively small, we use straight lines in the latitude-longitude space as an estimation, with negligible error. This approximation makes the computation faster, improving the overall performance of the algorithm.

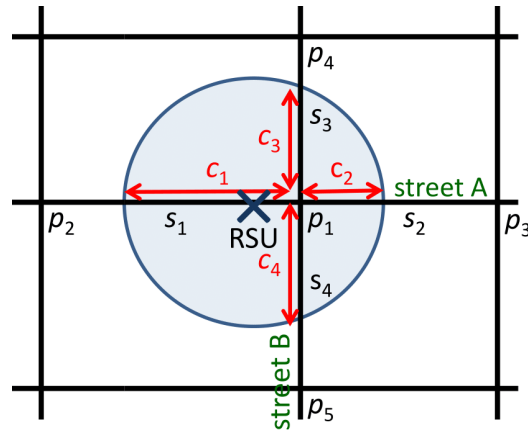


Figure 4. Calculation of the vehicles attended by a RSU.

In order to avoid situations where the fitness function computations take into account several times the same area (i.e., some vehicles in that area are counted multiple times), it is necessary to keep track of the number of vehicles that each installed RSU has already served. That number of vehicles depends on the type of the data streams that the RSU is providing (data, voice or video), since the minimum QoS required by each type of stream limits the maximum number of vehicles that can be simultaneously attended by a given RSU type.

6. EXPERIMENTAL ANALYSIS

This section presents the details of the experimental analysis performed to evaluate the proposed NSGA-II to solve the RSU-DP.

6.1. Development and execution platform

The proposed MOEA was implemented using the ECJ library, a Java-based evolutionary computation research system developed at ECLab Evolutionary Computation Laboratory, George Mason University.³⁵ ECJ includes easily modifiable classes for solving multiobjective optimization problems using the NSGA-II algorithm.

The experimental analysis was performed over an AMD Opteron 6172 2.10 GHz server, with 24 cores and 24 GB RAM at Cluster FING, the High Performance Computing facility at Universidad de la República, Uruguay.²⁷

Since computing the fitness of an individual is highly CPU-intensive, the evaluation of the population is performed in parallel using 24 Java-threads. Thus, each thread evaluates 3 individuals of the population.

For each problem instance (map, traffic, and application), we performed 30 independent executions of the proposed MOEA. For the heuristics used as a baseline for the results comparison, we performed one execution for the PageRank algorithm (deterministic) and 30 independent executions for the Knapsack algorithm (randomized). Including the parameter setting and validation experiments, we performed a total number of **1080** executions of NSGA-II and **279** executions of the heuristic algorithms.

6.2. Problem instances

In order to evaluate the proposed MOEA, we defined a real world problem scenario that is relevant for our community. We included real information for a number of elements: a real map of the city of Málaga, real road traffic data from the Málaga Council, real RSU network interfaces/antennas, and real applications to be executed over the VANET. The main details about these elements are presented next.

6.2.1. Map Figure 5 shows the map of Málaga considered in the experimental analysis. The map covers an area of 42557 km² in the city, including a total number of 106 points, which define 128 segments with lengths between 55.5 and 1248.2 m, and an average length of 483.9 m. 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).



Figure 5. Segments defined over the real map of Málaga.

6.2.2. Road traffic data The traffic data used in our experiments are based on the information collected by the Málaga City Council using a set of sensors located along the roads. These sensors returned the total number of vehicles that circulated during the first six months of 2015. The information is publicly available at the Málaga Council Mobility website.²⁶

We used the traffic information to define the *normal* traffic pattern in our RSU-DP scenario. In addition, we applied two probabilistic multiplicative factors over the *normal* pattern, to define a *low* pattern, reducing the traffic randomly in [0–20%] and a *high* pattern, increasing the traffic randomly in [0–20%]. These patterns represent situations with low and high road traffic density, respectively, according to the real data from the Málaga City Council (in fact, studying the traffic statistics for peak hours provided by the Council, we verified that that the number of vehicles for all main roads are about 20% higher than in a normal traffic scenario).

6.2.3. RSUs The initial effort to standardize the DSRC radio technology took place in the ASTM 2313 working group in the U.S. This effort migrated to the IEEE 802.11 standard group that proposed IEEE 802.11p, which stands for *wireless access in vehicular environments* (WAVE).¹⁸ Following these specifications, the RSUs considered in our study are equipped with a real world IEEE 802.11p commercial network interface. This hardware configures a realistic scenario and allows computing useful results from the point of view of both the research and technological communities.

Each network interface is connected to an external antenna to improve the communication capabilities, 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 the radio range that can be obtained, and the better the QoS provided by the infrastructure.

The RSUs analyzed in the problem instances solved in the experimental analysis differ in the antenna connected. Three different commercial omni-directional antennas, which can be found in online shops (e.g., Cetacea⁷), are considered. The antennas differ in the gain offered and the cost. A summary of the main features of such antennas is presented in Table II. Our study does not exclude the possibility of incorporating new communication devices (network interfaces and antennas) for the city infrastructure. The proposed algorithms do not depend on the type or features of the RSUs considered.

Table II. General information about the used antennas to define different RSU.

<i>type</i>	<i>commercial model</i>	<i>gain</i>	<i>cost</i>
t_1	Echo Series Omni Site 6dBi	6 dBi	\$121.70
t_2	Echo Series Omni Site 9dBi	9 dBi	\$139.20
t_3	Echo Series Omni Site 12dBi	12 dBi	\$227.50

One of the main features that defines a given RSU is its *effective radio range* (ERR). EER indicates the farthest distance at which the RSU may exchange data packets with the vehicles, and it is a relevant metric to evaluate the performance of the communications provided by the VANET infrastructure. In fact, ERR is one of the components we evaluate in the fitness function to compute the QoS provided by each RSU deployed in the studied scenario.

To determine the ERR metric for each studied RSU, we performed realistic VANET simulations evaluating the PDR at different distances (from 0 to 650 m). The experiments were performed by using the ns-2 simulator²⁹ to evaluate the communications. The simulated VANET scenario is defined by a given RSU and 10 moving cars at 40 Km/h (11.11 m/s) that utilize IEEE 802.11p network devices in a urban area. During the simulations, the RSU sent continuous data streams at 256 Kbps to the vehicles. The probabilistic Nakagami radio propagation model³² was used to represent the channel fading characteristics of urban scenarios. Each scenario was simulated 15 times to obtain robust average PDR values. In order to ensure a realistic QoS that guarantees reliable communications, we defined the ERR of each RSU as the distance at which the average PDR is equal or higher than 66.67% (i.e., less than one packet lost for each three packets transmitted).

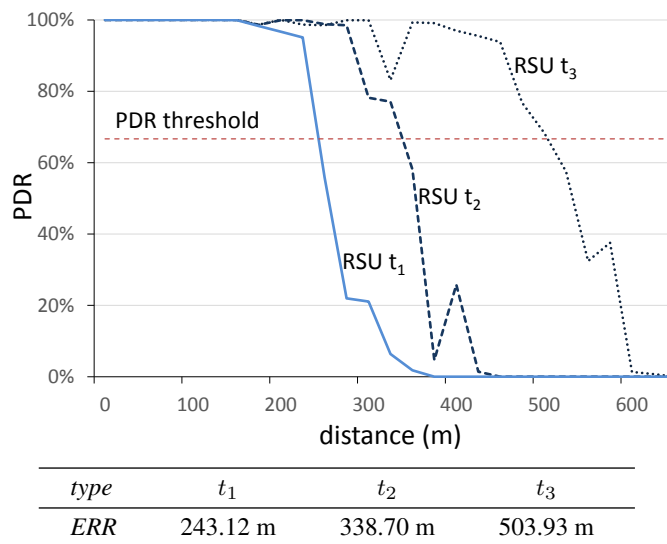


Figure 6. ERR experimental results.

Figure 6 reports the experimental results of the simulations to obtain the ERR for each RSU type. According to the PDR threshold defined, the ERR for the RSU type t_1 is 243.12 m, for RSU type t_2 is 338.70 m, and for RSU type t_3 is 503.93 m (see the values at the bottom of Figure 6).

6.2.4. Applications As we introduced in Section 1, VANET communications comprise mainly three types of applications: road safety, traffic efficiency, and infotainment. The two first types of applications rely on the exchange of small data packets with very short communication delays. The infotainment applications comprise a wide variety of applications that are principally based on receiving audio/voice and video streams.^{5,17}

One of the main contributions of the study reported in this article is that we defined problem instances taking into account the requirements of the different types of VANET applications. This analysis allows the designers to prioritize the applications' constraints in the final RSU deployment. Thus, for each RSU type, the computed QoS metric takes into account the maximum number of vehicles that can be served while fulfilling the requirements for each VANET application type (MU). The QoS constraints for each VANET application type evaluated in this study are summarized in Table III, based on the study by Chantaksinopas et al.⁸

Table III. QoS requirements of VANET applications taken into account in this study ($Q(A_i)$).

<i>application type</i>	<i>packet size (bytes)</i>	<i>generated data flow</i>	<i>QoS requirements</i>
data (A_1)	238 bytes	19 kbps (10 packets/s)	E2ED<100 ms & PDR=100%
voice/audio (A_2)	238 bytes	25 kbps	E2ED<400 ms & PDR>16%
video (A_3)	791 bytes	384 kbps	E2ED<400 ms & PDR>8.33%

In order to evaluate the MU function described in Section 2, we defined an iterative procedure based on performing realistic urban VANET simulations for each RSU (R_i) and application (A_j).

The procedure consists in simulating a given scenario that includes a RSU of a given type R_i and different number of vehicles spread through a circular area with radio $ERR(R_i)$. In these simulations, the VANET nodes generate data flows (traffic) according to the A_i application (see the third column on Table III). Therefore, the VANET scenario is defined according three different parameters (n, R_i, A_j).

The evaluation procedure starts by simulating the scenario with one vehicle ($1, R_i, A_j$) and computing the two relevant QoS metrics defined for the problem (PDR and E2ED). After that, a new vehicle is added and the new configuration is simulated. The iterative method stops when the computed QoS metrics do not accomplish the requirements defined in $Q(A_i)$ function. Finally, $MU(R_i, A_j)$ is the number of vehicles previous to the last one that was simulated. The computed results for each RSU and application type are summarized in Table IV.

Table IV. Number of vehicles that can be served for each RSU and application types.

<i>RSU type</i>	<i>application type</i>		
	safety	voice/audio	video
t_1	45	34	31
t_2	45	44	34
t_3	46	52	37

According to the results presented in Table IV, for example, all RSUs equipped with an antenna type t_2 can provide an adequate QoS to 34 vehicles executing a video stream application, and so on.

6.3. Heuristic methods used as a baseline for the comparison

In order to evaluate the quality of the solutions computed by the proposed NSGA-II, we implemented two versions of well-known heuristic methods to solve a different variant of the

RSU location problem, originally proposed by Ben Brahim et. al.⁴ the PageRank heuristic and the Knapsack algorithm.

6.3.1. Constructive PageRank heuristic PageRank is a voting algorithm, initially developed to compute the importance of web pages in the Internet by taking into account the number of inbound and outbound links from and to other web pages.¹⁹

The PageRank algorithm has been previously applied to solve the RSU deployment problem by Ben Brahim et al.⁴ In that study, the authors applied the PageRank version for weighted graphs to rank the potential locations for RSUs (road intersections) according to mobility-related information (e.g., traffic density, average speed of vehicles). The road traffic network is modeled as a graph with weighted links, which represent roads (segments), and vertices, which represent the intersections. The weight of each link is given by mobility-related information (e.g., density, average speed).

The *weighted PageRank* is applied to a given directed graph $G = (V, E)$ defined by a set of vertices V , and a set of edges E . The algorithm starts by setting the PageRank value of all vertices v_i to a fixed value d : $PR^W(v_i) = d, \forall v_i \in V$. d is known as the *dumping parameter* and its default value is 0.85. Then, an iterative process is performed until a stop condition is reached (the convergence value is below a given threshold or a maximum number of iterations performed). In this iterative process, for a given vertex v_i , $PR^W(v_i)$ is computed by

$$PR^W(v_i) = (1 - d) + d \times \left(\sum_{v_j \in In(v_i)} w_{ij} \times \frac{PR^W(v_j)}{\sum_{v_k \in Out(v_j)} w_{jk}} \right) \quad (3)$$

where $In(v_i)$ is the set of vertices that point to it (*predecessors*), and $Out(v_i)$ is the set of vertices that v_i points to (*successors*), and w_{ij} is the weight that for the edge that connects v_i and v_j .

In our study, we consider all the road segments as potential locations to install the RSUs, and not just the intersections as proposed by Ben Brahim et al.⁴ Therefore, we adapt the weighted PageRank algorithm with the purpose of sorting the segments (the edges of the graph) according to the rank value, and after that, applying a constructive heuristic over the sorted vector of segments.

The weighted graph $G = (V, E)$ is defined by the set of points P and the set of segments S in the RSU-DP formulation ($V=P$ and $E=S$); The weight of each edge w_{jk} is given by the weight of the represented segment $W(s_i)$, $s_i = (p_j, p_k)$, defined by Equation 4.

$$w_{jk} = W(s_i) = NV(s_i) \times \frac{len(s_i)}{sp(s_i)} \quad (4)$$

The rank value for each segment $s_i = (p_j, p_k)$ is computed as the sum of the PageRank values of p_j and p_k , i.e., $SR^W(s_i) = PR^W(p_j) + PR^W(p_k)$. Thus, the segments are ranked in a sorted vector S^{PR} in which $s_i^{PR}, s_j^{PR} \in S, i < j \Leftrightarrow SR^W(s_i^{PR}) > SR^W(s_j^{PR})$.

Once the segments are sorted in S^{PR} , a constructive heuristic is applied to select and locate the RSUs. The heuristic iterates over the sorted vector S^{PR} starting by the first segment (s_1^{PR}), which is the best ranked one. For each segment $s_i^{PR} \in S^{PR}$, the different QoS provided for each one of the three RSU types when installing them in one of 10 different equidistant points within the segment that are located in positions $n \times 0.1 \times len(s_i^{PR}), n \in [0, 9]$, are computed. Thus, the constructive algorithm evaluates the QoS of 30 different possible configurations (3 RSU types \times 10 locations) in each segment. It considers the configuration (RSU type and location) that provides the best QoS, if the whole QoS increases at least 1% regarding the previous segment. Otherwise, the constructive PageRank heuristic does not locate any RSU in the current segment s_i^{PR} .

6.3.2. Randomized Knapsack algorithm The 0-1 Knapsack problem²² is a well-known optimization problem that assumes having a bag with a capacity W and a set of objects characterized by their benefit value p_i and their weight w_i . The goal of the 0-1 Knapsack problem is to find the optimal subset of objects to include in the bag, maximizing the benefit P without exceeding the capacity W .

The RSU-DP problem can be reduced to a 0-1 Knapsack problem, which can be solved applying a dynamic programming algorithm. In this work, we adapt the Knapsack algorithm by Ben Brahim

et. al⁴ to solve the RSU-DP following a non-deterministic dynamic programming approach, the *Randomized Knapsack* (RandKS) presented in Algorithm 3.

Algorithm 3 Schema of the RandKS($B, SS, C, n, Ksol$).

```

1: if ( $n==0$  or  $B \leq 0$ ) then
2:    $Ksol \leftarrow \emptyset$  ▷ No more segments or budget
3:   return 0
4: else
5:   for  $k \leftarrow 1$  to  $K$  do ▷ For all RSU types
6:     if  $B < C(t_k)$  then ▷ No budget enough
7:        $cov_k = \text{RandKS}(B, SS, C, n-1, Ksol)$ 
8:     else
9:        $loc \leftarrow \text{random} \in [0, 1)$ 
10:       $cov_k \leftarrow \text{coverage}((s_n, t_k, loc), Ksol) + \text{RandKS}(B-C(t_k), SS, C, n-1, Ksol)$ 
11:    end if
12:     $rsu\_index \leftarrow \text{getIndex}(\max(cov_i))$ 
13:    if not  $rsu\_index==0$  then
14:       $Ksol \leftarrow Ksol \cup (s_n, t_k, location)$ 
15:    end if
16:  end for
17: end if

```

The Randomized Knapsack algorithm defines the set SS , which stores all the possible pairs of road segments and RSU types ($S \times R$), i.e., $SS = \{(s_1, t_1), (s_1, t_2), \dots, (s_1, t_K), \dots, (s_n, t_K)\}$. The elements of SS are the ones to include in the Knapsack bag when building a RSU infrastructure for a VANET. The final solution is stored in the set $Ksol$, which stores tuples that include information about the installed RSUs: the road segment s_i , the RSU type t_k , and the *location* of the RSU in the segment, i.e., $Ksol$ includes $(s_i, t_k, location)$ tuples.

The *location* loc of the RSU in the segment is a real number in the range $[0, 1)$, that represents the relative value of the position in the segment, as explained for the solution encoding in NSGA-II in Section 5.1. In the original (deterministic) Knapsack algorithm by Ben Brahim et. al,⁴ the location of RSU is limited to the corners of the streets, i.e., the extremes of each segment. In our Randomized Knapsack algorithm, the location of the RSU is picked randomly within each segment, because we are working with a set of infinite possible locations, which cannot be explored one by one.

Two new functions are defined for the RandKS algorithm: i) $\text{coverage}((s_n, t_k, loc), sol)$, which computes the complete coverage provided by the RSUs stored in sol plus the RSU located in the segment s_n defined by (s_n, t_k, loc) ; and ii) $\text{getIndex}(\max(cov_i))$, which returns the index of the RSU type that obtained the best (maximum) coverage.

6.4. Multiobjective optimization metrics

A large number of metrics have been proposed in the literature to evaluate MOEAs.^{10,12} In this work, we apply two relevant metrics in order to evaluate the results obtained by the NSGA-II algorithm: hypervolume and relative hypervolume (RHV). These two metrics allows evaluating, in terms of both convergence and correct sampling, the set of non-dominated solutions of the problem.

The hypervolume measures the volume (in the objective functions space) covered by the computed Pareto front. The relative hypervolume is the ratio between the volumes (in the objective functions space) covered by the computed Pareto front and the true Pareto front. The ideal RHV value is 1.

The true Pareto front—which is unknown for the problem instances studied—is approximated by the set of non-dominated solutions found for each problem instance solved, in each one of the 30 execution of the proposed MOEA.^{10,12}

6.5. Parametric configuration

Due to the stochastic nature of EAs, a parameter configuration is mandatory prior to the experimental analysis. For this purpose, we performed an experimental analysis to configure two important parameters of the NSGA-II algorithm: the crossover probability (p_c) and the mutation probability (p_m).

For the parameter setting, we use a set of problem instances which is different from the real-world problem instance, in order to avoid bias in the experimental analysis. We considered a smaller region of the map of Málaga, comprised of 74 segments, and we defined 3 different instances over this region corresponding to each type of application (data, voice, and video). For the parameter configuration we considered the scenario with normal traffic. For each parameter, three candidate values were tested: $p_c \in \{0.5, 0.7, 0.9\}$, and $p_m \in \{0.1, 0.01, 0.001\}$.

We performed 30 independent executions of 10000 generations over each problem instance using each one of the different combinations of the candidate parameter values, thus totalling 810 executions of the proposed MOEA.

Table V reports the hypervolume achieved using each parameter configuration on the scenario involving a video application, which is representative of the set of scenarios used in the parameter tuning. The mean, median, and standard deviation (σ) are shown along with the minimum (*min*) and maximum (*max*) values achieved. Furthermore, the p -value corresponding to the Shapiro-Wilk test for normality is displayed as well as the *Friedman rank* values corresponding to each parameter configuration.

The results from the Shapiro-Wilk test did not allow to confidently state whether the results samples follow a normal distribution (in six out of nine instances, the p -value was larger than 0.05). Therefore, the non-parametric Friedman rank test was used in order to compare the different parameter configurations against each other. The p -values for the Friedman rank test did not allow to state whether there is one configuration that outperforms all the others with statistical significance. Therefore, we decided to use the configuration $p_C = 0.7$; $p_M = 0.1$, which achieved the best mean and median hypervolume values.

Table V. Parameter configuration results for the scenario with a video application.

p_c	p_m	hypervolume ($\times 10^6$)					p -value S-W	<i>Friedman</i> <i>Rank</i>
		<i>mean</i>	<i>median</i>	σ	<i>min</i>	<i>max</i>		
0.5	0.001	9.27	9.27	0.04	9.18	9.36	0.99	148.00
	0.01	9.27	9.28	0.05	9.16	9.36	0.86	160.00
	0.1	9.26	9.27	0.05	9.11	9.34	0.04	147.00
0.7	0.001	9.27	9.29	0.06	9.14	9.38	0.15	143.00
	0.01	9.28	9.27	0.05	9.21	9.40	0.15	149.00
	0.1	9.29	9.30	0.05	9.20	9.38	0.76	171.00
0.9	0.001	9.25	9.26	0.07	9.07	9.36	0.34	134.00
	0.01	9.25	9.28	0.06	9.10	9.34	0.01	134.00
	0.1	9.28	9.28	0.04	9.14	9.35	0.03	164.00

6.6. Numerical results

This subsection reports the numerical results achieved in the experimental evaluation of the proposed NSGA-II algorithm for the RSU-DP. The results shown are those corresponding to the 30 independent executions performed on each of the 9 problem instances studied: 3 different applications (data, voice, video), each with 3 different traffic levels (normal, high, low) for the proposed NSGA-II and the baseline heuristics (PageRank/Knapsack).

The RHV metric is used to compare the solutions computed by NSGA-II against the one computed using the baseline heuristics. RHV is a good indicator of both convergence towards an ideal Pareto front and diversity among the set of non-dominated solutions. The ideal Pareto front for a given problem instance is approximated with the set of non-dominated solutions computed by all algorithms on every independent execution of that instance.

In order to compare the RHV values achieved by each algorithm we used two statistical tests. We first performed the Shapiro-Wilk test to assess normality over the RHV values obtained by each algorithm on each problem instance. In seven out of nine instances, the results from the Shapiro-Wilk test did not allow to state whether the samples follow a normal distribution or not. Therefore, the Friedman Rank test was used to assess whether the results achieved by one algorithm outperformed the others.

Table VI reports the RHV values achieved by NSGA-II, Knapsack and PageRank on all 9 problem instances. The *mean*, standard deviation (σ), and best (*max*) values were computed over the 30 independent executions performed for each algorithm. In addition, the table also reports the *p*-value from the Shapiro-Wilk test (*S-W*), the rank from the Friedman test (*Rank*), and the *p*-value from the Friedman test.

Table VI. Relative hypervolume values achieved by NSGA-II, Knapsack, and PageRank.

		normal			high			low		
		data	voice	video	data	voice	video	data	voice	video
<i>max</i>	NSGA-II	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Knapsack	0.82	0.85	0.95	0.8	0.84	0.96	0.83	0.86	0.95
	PageRank	0.73	0.82	0.73	0.7	0.81	0.71	0.76	0.83	0.74
<i>mean</i>	NSGA-II	0.98	0.98	0.98	0.99	0.95	0.98	0.99	0.99	0.98
	Knapsack	0.8	0.84	0.94	0.79	0.82	0.94	0.82	0.85	0.94
	PageRank	0.73	0.82	0.73	0.7	0.81	0.71	0.76	0.83	0.74
$\sigma (\times 10^{-3})$	NSGA-II	3.72	2.83	3.00	2.73	176	3.79	3.74	3.65	3.33
	Knapsack	5.57	4.57	7.77	5.88	6.81	6.43	7.26	6.53	8.05
	PageRank	0	0	0	0	0	0	0	0	0
<i>S-W</i>	NSGA-II	0.93	0.17	0.6	0.1	0	0.25	0.3	0.11	0.35
	Knapsack	0.49	0.91	0.6	0.29	0.88	0.94	0.33	0.37	0.15
	PageRank	1	1	1	1	1	1	1	1	1
<i>Rank</i>	NSGA-II	90	90	90	90	88	90	90	90	90
	Knapsack	60	60	60	60	60	60	60	60	60
	PageRank	30	30	30	30	32	30	30	30	30
<i>p</i> -value Friedman		$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$

Table VII. NSGA-II improvements over Knapsack and PageRank algorithms.

		improvement in QoS (%)				improvement in cost (%)	
		cost=\$10000		cost=\$15000		QoS=2500	
		Knapsack	Pagerank	Knapsack	Pagerank	Knapsack	Pagerank
normal	data	22.37	44.89	-	-	31.37	39.48
	voice	19.06	29.76	-	-	25.34	28.93
	video	2.58	49.97	5.97	24.63	6.46	31.18
high	data	24.68	52.71	-	-	30.17	42.4
	voice	22.11	33.82	-	-	22.35	28.26
	video	0.83	52.55	6.42	34.13	1.84	34.18
low	data	18.44	35.61	-	-	34.09	33.52
	voice	17.83	25.78	-	-	33.75	25.51
	video	6.91	46.89	5.04	14.03	15.06	28.53

The results in Table VI demonstrate that NSGA-II is able to achieve significantly better RHV values than the two baseline heuristics adapted from the literature, both in average and in the best case. The Friedman rank test allows to state with statistical confidence that NSGA-II is able to

outperform both Knapsack and PageRank algorithms in all problem instances (p -value $< 10^{-6}$ in all comparisons). This fact suggests that NSGA-II accurately computes fronts that converge towards an ideal Pareto front of the problem while simultaneously maintaining diversity among the set of non-dominated solutions.

Table VII reports the improvements achieved by NSGA-II over Knapsack and PageRank heuristics. The reported improvements are those achieved when comparing specific (realistic) solutions from the global Pareto fronts computed by each algorithm (i.e., combining the results of all 30 independent executions performed), as usual in the MOEA literature.^{10,12}

In Table VII, the improvements regarding each problem objective are evaluated by comparing the values of QoS/cost computed by each studied algorithm when considering a fixed cost/QoS used as a reference value, respectively. The improvements in QoS are measured comparing the values of QoS achieved by each algorithm with a fixed cost of \$10000 (all cost values are expressed in US dollars). This cost is a reasonable value for the budget to invest in order to deploy a RSU infrastructure for VANET in a city scaled area, just like in the case of Málaga. In addition, in the scenarios corresponding to video applications, the QoS improvements with a fixed cost of \$15000 are also reported, since the deployment costs for infrastructures that support this type of applications tend to be more expensive. Following a similar approach, the improvements in cost are reported comparing the cost values of solutions achieved by each algorithm with a fixed QoS of 2500.

In any case, since the proposed algorithm solves the problem following a multiobjective approach, a human decision-maker will always have the last word about the solution to implement: the decision-maker from the City Council will be in charge of selecting one solution of the set of non-dominated ones for the real implementation over the city.

The results in Table VII clearly state that NSGA-II is able to improve the solutions computed by both Knapsack and PageRank algorithms in all scenarios. In terms of QoS, the maximum improvements are of 24.68% over Knapsack and 52.71% over PageRank, both in the instance with high traffic and a data application. The improvements of costs are of up to 34.09% over Knapsack (instance with low traffic and a data application) and 39.48% over PageRank (instance with normal traffic and a data application).

In order to provide a better insight into the main advantages of the proposed NSGA-II algorithm, Figure 7 shows the global Pareto fronts achieved by each algorithm on each problem instance. It can be observed that NSGA-II is able to compute accurate Pareto fronts with good convergence and diversity properties. The improvements of NSGA-II on both QoS and cost are clear for data and voice applications, in all three traffic patterns studied. When considering a more demanding application (video), the improvements of NSGA-II over Knapsack are observed for large values of QoS (i.e., $QoS > 2500$). The PageRank algorithm was consistently the worst method to solve the RSU-DP in all problem instance studied.

7. CONCLUSIONS AND FUTURE WORK

This article reports the application of a multiobjective evolutionary approach to solve the problem of locating roadside infrastructure for vehicular networks over realistic urban areas.

A multiobjective formulation of the problem was introduced, considering the QoS and cost objectives. A specific NSGA-II evolutionary algorithm was designed, by including a problem-related encoding and ad-hoc mutation operators to explore the set of possible locations. A parallel model was applied in order to efficiently perform the evaluation of solutions in the proposed MOEA.

The NSGA-II algorithm was evaluated on a city-scaled area using the real map and up-to-date (year 2015) traffic information from the city of Málaga, in Spain. The problem instances were built considering three different types of commercial antennas and a set of realistic VANET parameterizations that could ease the use of demanding applications related to data, voice, and video communications. The scenarios used for the experimental evaluation considered three different traffic patterns and three different types of applications, each one with different infrastructure requirements.

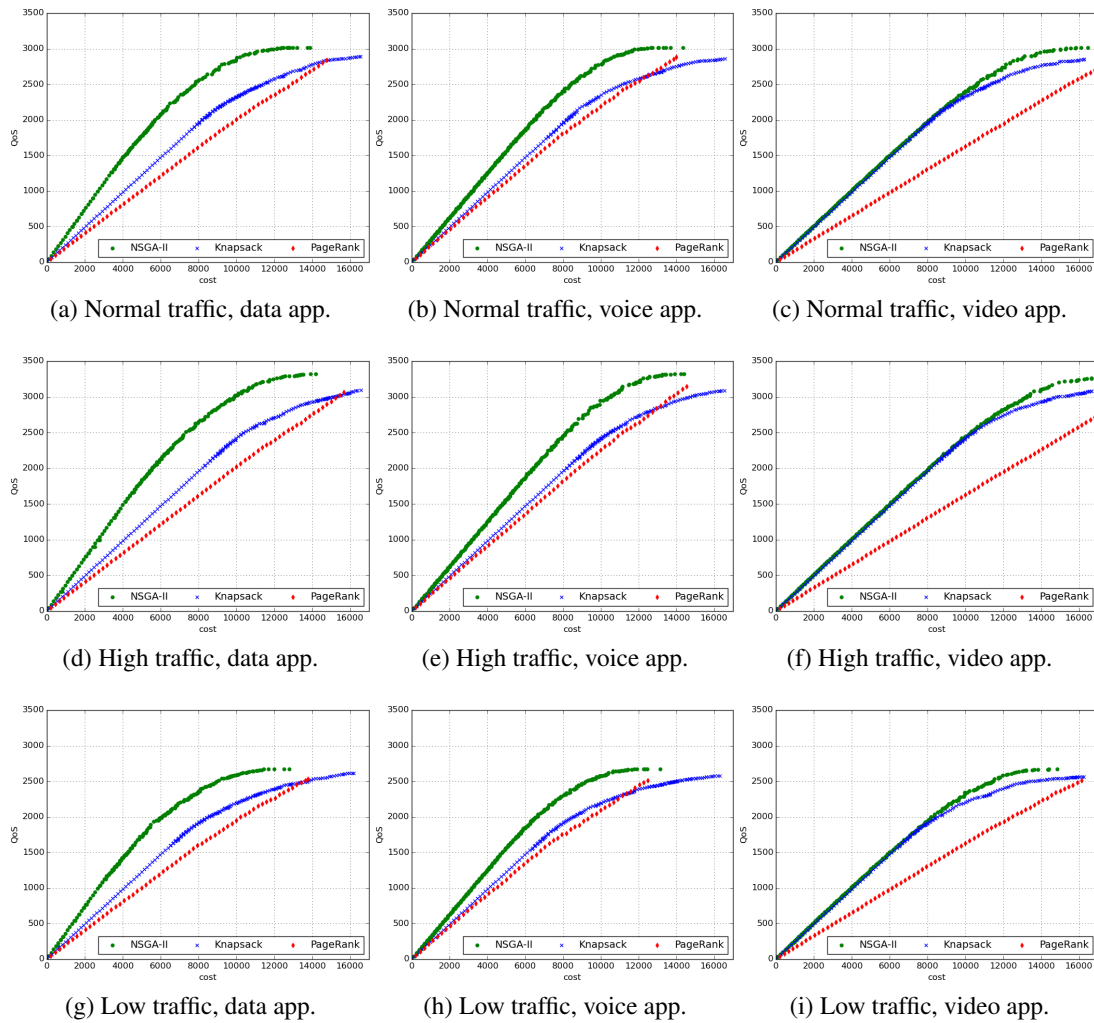


Figure 7. Global Pareto fronts.

Two heuristics were implemented to solve the problem, based on related works from the literature: a Randomized Knapsack algorithm and a constructive PageRank heuristic. These traditional methods for RSU planning were used as a reference baseline to compare the solutions computed by the proposed NSGA-II algorithm. In the experiments performed, the proposed MOEA has shown good problem solving capabilities, computing accurate Pareto fronts for the problem. NSGA-II also allows improving over the two baseline heuristics, regarding the multiobjective optimization metrics evaluated and the two problem objectives.

According to the results from the experimental analysis, NSGA-II was able to consistently compute better results than the baseline heuristics. In the best case, NSGA-II was able to outperform the Randomized Knapsack algorithm up to 24.68% and the PageRank heuristic up to 52.71% in terms of QoS. The improvements in cost were up to 34.09% and 39.48% over Knapsack and PageRank, correspondingly. The computed Pareto fronts indicate that NSGA-II provides better and more robust solutions to the problem, especially for those scenarios considering data and voice applications. The experimental analysis also demonstrates that when considering a more realistic scenario (including real applications, real traffic information, etc.) the evolutionary approach is able to achieve larger improvements over the heuristics than those reported in previous related works.

The main lines for future work are related to extend the experimental analysis to consider other real areas from cities in other countries. Right now, we are working on building a real RSU-DP scenario for VANETs in the city of Montevideo, Uruguay, using real data from the local authorities and GPS information from the public transport. We also plan to extend the problem formulation to consider additional information from important events in vehicular networks (such as accidents, traffic jams, etc.) in order to model a more realistic scenario for the problem.

ACKNOWLEDGMENTS

The work of R. Massobrio and S. Nesmachnow has been partially funded by ANII and PEDECIBA, Uruguay. The work of J. Toutouh and E. Alba has been partially funded by the Spanish MINECO and FEDER project TIN2014-57341-R.

CONTRIBUTIONS

Renzo Massobrio and Jamal Toutouh carried out most of the experimental study, including programming the multiobjective evolutionary approach and the heuristics to solve the problem, and performing the VANET experiments to build the real scenario studied in the experimental analysis. They also participated in the writing of the paper. Sergio Nesmachnow worked in the design and implementation of the software methods, the design and evaluation of the case study in Málaga, and is responsible of the writing of the paper. Enrique Alba has contributed in the revision of the manuscript and writing of the paper. The international cooperation was coordinated by S. Nesmachnow (U. de la República) and J. Toutouh (U. of Málaga). S. Nesmachnow and E. Alba were responsible for the funds dedicated to the international cooperation.

REFERENCES

1. E. Alba, G. Luque, and S. Nesmachnow. Parallel metaheuristics: Recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.
2. B. Aslam, F. Amjad, and C. Zou. Optimal roadside units placement in urban areas for vehicular networks. In *IEEE Symposium on Computers and Communications*, pages 423–429, July 2012.
3. T. Bäck, D. Fogel, and Z. Michalewicz, editors. *Handbook of evolutionary computation*. Oxford University Press, 1997.
4. Mohamed Ben Brahim, Wassim Drira, and Fethi Filali. Roadside units placement within city-scaled area in vehicular ad-hoc networks. In *3rd International Conference on Connected Vehicles and Expo*, pages 1–7. IEEE, 2014.
5. C. Campolo, A. Molinaro, and R. Scopigno, editors. *Vehicular ad hoc Networks - Standards, Solutions, and Research*. Springer, 2015.
6. E. Cavalcante, A. Aquino, G. Pappa, and A. Loureiro. Roadside unit deployment for information dissemination in a VANET: An evolutionary approach. In *14th Genetic and Evolutionary Computation Conference*, pages 27–34, 2012.
7. Cetacea Wireless Solutions Company shop. Online <https://shop.cetacea.com/>. Retrieved December 2015.
8. I. Chantaksinopas, W. Lee, A. Prayote, and P. Oothongsap. Delay-sensitive applications in VANET and seamless connectivity: The limitation of UMTS network. *International Journal Computer Science & Network Security*, 12:54–61, 2012.
9. H. Cheng, X. Fei, A. Boukerche, A. Mammeri, and M. Almula. A geometry-based coverage strategy over urban VANETs. In *10th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pages 121–128, 2013.
10. C. Coello, D. Van Veldhuizen, and G. Lamont. *Evolutionary algorithms for solving multi-objective problems*. Kluwer, New York, 2002.
11. M. Deakin. *Smart Cities: Governing, Modelling, and Analysing the Transition*. Routledge, 2013.
12. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. J. Wiley & Sons, Chichester, 2001.
13. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
14. J. Falocchio and H. Levinson. *Road Traffic Congestion: A Concise Guide*. Springer Tracts on Transportation and Traffic. Springer International, 2015.
15. F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.
16. D. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley, New York, 1989.

17. H. Hartenstein, K. Laberteaux, and I. Ebrary. *VANET: vehicular applications and inter-networking technologies*. Wiley Online Library, 2010.
18. D. Jiang and L. Delgrossi. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In *IEEE Vehicular Technology Conference*, pages 2036–2040, 2008.
19. A. Langville and C. Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.
20. Y. Liang, H. Liu, and D. Rajan. Optimal placement and configuration of roadside units in vehicular networks. In *IEEE 75th Vehicular Technology Conference*, pages 1–6. IEEE, 2012.
21. C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve. Data aggregation and roadside unit placement for a VANET traffic information system. In *5th ACM International Workshop on Vehicular Inter-Networking*, pages 58–65, 2008.
22. S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.
23. R. Massobrio, S. Bertinat, S. Neschachnow, J. Toutouh, and E. Alba. Smart placement of RSU for vehicular networks using multiobjective evolutionary algorithms. In *Latin America Conference on Computational Intelligence*, pages 1–6, 2015.
24. R. Massobrio, J. Toutouh, and S. Neschachnow. A multiobjective evolutionary algorithm for infrastructure location in vehicular networks. In *7th European Symposium on Computational Intelligence and Mathematics*, pages 1–6, 2015.
25. S. Mendes, J. Gómez-Pulido, M. Vega-Rodríguez, J. Sánchez-Pérez, Y. Sáez, and P. Isasi. The radio network design optimization problem. In *Biologically-Inspired Optimisation Methods*, pages 219–260. Springer Science + Business Media, 2009.
26. Área de Movilidad, Ayuntamiento de Málaga. Online <http://movilidad.malaga.eu/>. Retrieved December 2015.
27. S. Neschachnow. Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República. *Revista de la Asociación de Ingenieros del Uruguay*, 61:12–15, 2010.
28. S. Neschachnow. An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*, 3(4):320–347, 2014.
29. The Network Simulator ns-2. Online <http://www.isi.edu/nsnam/ns>. Retrieved December 2015.
30. P. Patil and A. Gokhale. Voronoi-based placement of road-side units to improve dynamic resource management in VANETs. In *International Conference on Collaboration Technologies and Systems*, pages 389–396, 2013.
31. A. Reis, S. Sargento, F. Neves, and O. Tonguz. Deploying roadside units in sparse vehicular networks: What really works and what does not. *IEEE Transactions on Vehicular Technology*, 63(6):2794–2806, 2014.
32. S. Saunders and A. Aragon. *Antennas and Propagation for Wireless Communication Systems*. Wiley, New York, NY, USA, 1999.
33. O. Trullols, M. Fiore, C. Casetti, C. Chiasserini, and J. Ordinas. Planning roadside infrastructure for information dissemination in intelligent transportation systems. *Computer Communications*, 33(4):432–442, 2010.
34. C. Wang, X. Li, F. Li, and H. Lu. A mobility clustering-based roadside units deployment for VANET. In *16th Asia-Pacific Network Operations and Management Symposium*, pages 1–6, 2014.
35. D. White. Software review: the ECJ toolkit. *Genetic Programming and Evolvable Machines*, 13(1):65–67, 2011.