Renzo Massobrio, Jamal Toutouh, Sergio Nesmachnow

Multiobjective evolutionary algorithms for smart placement of roadside units in vehicular networks

Chapter 1

Multiobjective evolutionary algorithms for smart placement of roadside units in vehicular networks

1.1 Introduction

In the last decades, vehicular traffic has become a major concern in modern cities. This problem is mainly caused by a dramatic increase of the number of vehicles in the roads, which significantly impacts on mobility, road safety, travel times, and environmental issues [18]. Intelligent transportation systems (ITS) have risen from the combination of information and communication technologies (ICT) [17] to solve several problems related to vehicular planning, road mobility, etc. Among different techniques, ITS apply continuous information exchange and computational intelligence [32].

Vehicular ad hoc networks (VANETs) have emerged as one of the most promising solutions to provide network connectivity in vehicular environments to deploy ITS [6, 17]. VANETs are composed of a set of vehicles equipped with *on-board units* (OBU) and *roadside unit* (RSU) elements connected with each other using wireless technologies (see Figure 1.1). Specifically, VANET nodes use direct short range communications (DSRC) to continuously exchange relevant road traffic information with each other. However, the high mobility of the nodes results in frequent changes to the network topology as well as network disconnections, limiting the quality-of-service (QoS) of the system.

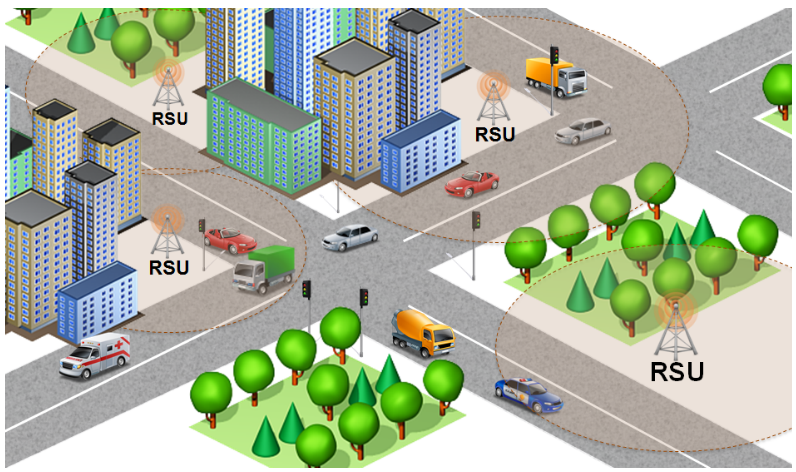


FIGURE 1.1: Basic VANET based ITS deployed in a city.

There are several VANET services (security, routing, information, entertainment) that use RSUs as information source or receiver. In addition, as RSU devices have better network capabilities than OBUs, they are used to extend the effective network coverage of the OBUs acting as multi-hop relays. Therefore, installing an efficient RSUs infrastructure along the roads (as a VANET backbone) may enhance the QoS of the network.

In this study, we focus on the efficient design of the RSU infrastructure, also known as *RSU Deployment Problem* (RSU-DP), which consists in selecting the best locations and RSU types in order to optimize both the service provided by the fixed infrastructure and the economical deployment costs. This is a tractable problem when dealing with small sized areas, but it results in a hard-to-solve problem for city-scaled instances, as the number of possible solutions becomes very large [33].

In the recent literature there are several studies applying heuristics and metaheuristics to address the efficient design of RSU infrastructures for VANETs [1, 7, 35]. Heuristics and metaheuristics allow traffic engineers to compute competitive solutions (RSUs deployments) in reasonable execution times. They apply soft computing search approaches, which are able to tolerate imprecision in data and handle partial solutions in order to build approximate solutions that satisfy the needs of the designers [27]. In this study, we propose applying *multiobjective evolutionary algorithms* (MOEAs) [10, 11] to design an RSU infrastructure to provide VANET services within a city-scaled road network in Málaga, Spain. In order to obtain realistic results, the experimental analysis is performed using real information about road traffic (road map and traffic flow) and hardware (network capabilities and costs).

Three MOEAs are studied and compared to solve the case of study presented in this chapter: a linear aggregation approach and the well-known NSGA-II and SPEA2 algorithms. An important aspect when solving optimization problems using MOEAs is the evaluation of the problem objectives. One of the objectives of the RSU-DP is to minimize the economical cost of the deployment. This objective is evaluated by computing the sum of the cost of each RSU, which depends on its specific type. The other objective in the RSU-DP is to maximize the network service provided by the RSU platform. In this study, a specific QoS model is proposed, considering the number of vehicles, speed, and coverage of street segments in the city, and a Monte-Carlo simulation approach is used to compute the corresponding QoS metric.

Thus, the main contributions of the research reported in this chapter are: *i*) introducing a multiobjective formulation of the infrastructure placement problem for VANETs by considering the optimization of the deployment cost and the QoS offered to the users, and *ii*) an experimental analysis of three MOEAs to solve a specific instance of the problem, built by using real data from the city of Málaga.

The rest of the chapter is organized as follows. Section 1.2 provides an overview about vehicular networks and related technologies. Section 1.3 describes the computational intelligence methods used in this study, introduces the multiobjective optimization concept, and describes the three MOEAs applied to solve the RSU-DP. Section 1.4 presents and formulates the multiobjective RSU-DP addressed in this work and reviews related studies solving different versions of the RSU infrastructure design problem. Section 1.5 details the specific implementation of the evaluated MOEAs to efficiently tackle the RSU-DP. Section 1.6 describes the greedy algorithms used as baseline RSU-DP solvers for comparing the results obtained by the proposed MOEAs, and reports the experimental analysis of the proposed methods on a set of different instances defined using real world data (Málaga road map and RSU hardware specification). Finally, Section 1.7 states the conclusions and describes the main lines for future work.

1.2 Vehicular Communication Networks

During the last decades, the use of ICT in vehicular environments has prompted the emergence of VANETs. VANETs are networks that connect vehicles equipped with OBUs and RSU with each other by using DSRC technologies [17]. Depending on the type of nodes involved in the communication, two cases are distinguished: *i*) *vehicle-to-vehicle* (V2V), when two vehicles communicate through ad hoc communications, and *ii*) *vehicle-to-roadside* (V2R), when the vehicles and roadside elements exchange data with each other.

Based on these types of communications, VANETs can provide a wide variety of powerful applications specifically targeted for vehicles. These application can improve the safety and efficiency for road users by gathering, processing, and broadcasting real time traffic information.

In the related literature, VANET applications are categorized into two groups: *safety* and *non-safety* applications. Safety applications use VANETs to exchange relevant information (such as speed, direction, and relative position) to reduce hazardous situations on the road, potentially reducing the number of car crashes. The U.S. Department of Transportation estimates that VANET safety applications have the potential to help drivers avoid or mitigate 80 percent of vehicle crashes [14]. Non-safety applications include a plethora of different services that enhance traffic efficiency (e.g., reduction of travel times or fuel saving), improve passengers' comfort, among others. Nevertheless, these groups are not completely orthogonal. For instance, an application designed to prevent road accidents also improves the efficiency, since it avoids a potential traffic jam the incident may cause.

The vehicular communication system is based in three main components (see Figure 1.2): OBUs, RSUs, and *application units* (AUs). These components are described below:

- OBUs are hardware devices integrated in the vehicles (VANET mobile nodes) to provide them with processing and communication features. Their main functions are: i) gathering and processing data collected from the sensors installed in the vehicle and ii) exchanging vehicular information with other VANET nodes (OBUs or RSUs) via IEEE 802.11p DSRC [13]. OBUs may include supplementary network interfaces, e.g., Bluetooth or cellular LTE.

- RSUs are hardware elements installed on the roadside infrastructure elements (e.g., traffic lights, road signals) and on specific dedicated VANET elements located along the roads. They include an IEEE 802.11p network interface to exchange information with other VANET nodes through DSRC. RSUs are also equipped with other network interfaces that provides them with connectivity to other types of networks, such as Internet. They play three major roles [5]: i) act as an information source or receiver in VANET applications (e.g., warning about the existence of roadworks); ii) extend the effective communication range by forwarding data to other VANET nodes (OBUs or RSUs) through multi-hop communications; and iii) provide Internet connectivity to OBUs.

- AUs may be either external devices connected to a given OBU, such as a smartphone, or a device integrated into the OBU forming a single physical unit. They are connected to the OBU through wired or wireless connection, such as Bluetooth. AUs provide the user interface that allow the user to interact with VANET applications and services.

According to the components involved in the communications, vehicular networks define three communication domains (see Figure 1.2):

- *In-vehicle domain*, which is defined by the OBU, AUs, and sensors connected in a given vehicle. Principally, the OBU uses the network links to gather data from the sensors and share the processed information with the AUs.
- *Ad hoc* or *V2X domain*, which is defined by the OBUs (vehicles) 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.
- *Infrastructure domain*, which may include the vehicles manufacturers, *trusted third parties* (TTP), service providers (SP), and trust authorities (TA). In the infrastructure domain, the RSU serves as a backbone bridge between the infrastructure environment and the ad hoc environment.

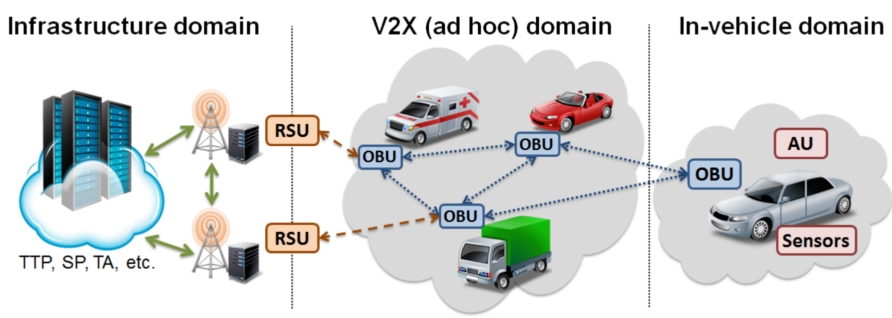


FIGURE 1.2: High-level architecture of VANETs.

In this study we focus on the installation of the required RSU infrastructure along the roads, which is an important problem to be addressed in order to successfully deploy VANETs services.

RSUs are important in VANETs because they operate as road traffic data repositories and information relays for different VANET safety applications.

Additionally, they are used to increase the effective coverage and robustness of the vehicular networks and can be also used as a gateway to the Internet, allowing data and context information to be collected, stored, and processed in upcoming Cloud infrastructures.

An example of a VANET application that uses RSUs (V2R communication) is the *Intersection Violation Warning (IVW)*, which warns the driver when the possibility of violating a red light is high [17]. A RSU installed in a traffic light controller broadcasts traffic light information including its location, light phase, light timing, and intersection geometry. Then, the OBU installed in the approaching vehicle compares this information with its trajectory and warns the driver if the signal violation is imminent (see Figure 1.3). In addition, the OBU sends to the surrounding VANET nodes a warning message indicating that there is a high probability for a hazardous situation to occur.

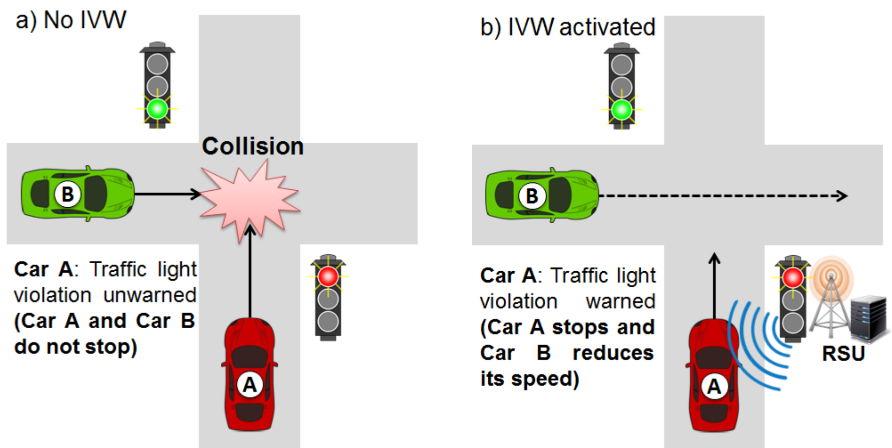


FIGURE 1.3: VANET warning on a intersection violation scenario: a) without IVW, car A runs the light and collides with car B; b) when IVW is applied, both cars are alerted and then car A stops and car B can reduce its speed.

1.3 Materials and methods: metaheuristics, evolutionary computation and multiobjective optimization

This section introduces the main concepts about the methods applied in our research: metaheuristics, evolutionary algorithms and multiobjective evolutionary algorithms.

1.3.1 Metaheuristics

Metaheuristics are high-level strategies to define algorithmic frameworks that allow designing efficient and accurate techniques to find approximate solutions for search, optimization, and learning problems [15]. They define high-level soft computing methods that can be applied to solve different optimization problems, by instantiating a generic resolution procedure, and needing relatively few modifications to be made in each specific case [27].

In practice, many optimization problems arising in nowadays real-world applications 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 manage very large volumes of data, or they are multimodal or multiobjective problems taking into account hard-to-evaluate optimization functions. This is the case for the problem solved in this chapter: the smart placement of roadside units for vehicular networks.

The smart placement of roadside units for vehicular networks is a special case of NP-hard problem, because it is a variant of the well-known Radio Network Design problem [25]. Thus, metaheuristics are efficient and accurate methods for solving realistic instances of the problem, that often cannot be solved in practice using exact optimization methods (e.g., enumerative search, branch and bound, dynamic/linear/integer programming) that are extremely time-consuming. In this article, we compare the application of multiobjective evolutionary metaheuristics to solve the proposed infrastructure placement problem. The main features of evolutionary algorithms, multiobjective optimization and multiobjective evolutionary algorithms are described next.

1.3.2 Evolutionary algorithms

Evolutionary algorithms (EAs) are non-deterministic metaheuristic methods that emulate the evolution of species in nature, in order to solve optimization, search, and learning problems [2, 16]. In the past thirty years, EAs have been successfully applied to solve optimization problems underlying many real and complex applications.

The generic schema of an EA is presented in Algorithm 1. An EA is an iterative technique (each iteration simulates a time step, and it 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 related to the objective function that evaluates how good a solution to the problem is. Every individual in the population represents a candidate solution for the problem, according to a specific encoding. The initial population is generated by applying 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, indicating its suitability to the problem (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), most notably including the *recombination* of parts from two individuals or random changes (*mutations*) in their contents, which are applied to 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 ← 0                                         {generation counter}
2: initialize(P(0))
3: while not stopcriterion do
4:     evaluate(P(t))
5:     parents ← selection(P(t))
6:     offspring ← variation operators(parents)
7:     P(t+1) ← replacement(offspring, P(t))
8:     t ← 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 shown in Algorithm 1 a GA defines selection, recombination and mutation operators, applying them to the population of potential solutions, which is replaced by the offspring 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.

1.3.3 Multiobjective optimization problems

Unlike traditional single-objective optimization problems, a Multiobjective Optimization Problem (MOP) proposes to optimize a group of functions, usually in conflict with each other. As a consequence, there is not a unique solution to the problem, but a *set* of solutions that represent different trade-offs among the optimizing functions values.

A generic formulation of a MOP is:

$$
\begin{array}{ll}
\text{min/max} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})) \\
\text{subject to} & \mathbf{G}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_j(\mathbf{x})) \geq 0 \\
& \mathbf{x} \in \Omega
\end{array}
$$

A MOP solution is a vector of decision variables $\mathbf{x} \in \Omega$ which satisfies the constraints formulated by the functions $\mathbf{G}(\mathbf{x})$, offering adequate trade-off values for the functions $f_i(\mathbf{x})$.

Considering a MOP that proposes to *minimize* a set of objective functions, a solution $\mathbf{w}$ is said to dominate the other solution $\mathbf{v}$ (it is denoted $w \prec v$), if $f_i(\mathbf{v}) \geq f_i(\mathbf{w}) \wedge \exists j / f_j(\mathbf{v}) > f_j(\mathbf{w})$. The set of optimal solutions for a MOP is composed by the non-dominated feasible vectors, named *Pareto optimal set*. It is defined as $P^* = \{\mathbf{x} \in \Omega / \nexists \mathbf{x}' \in \Omega, \mathbf{F}(\mathbf{x}') \preceq \mathbf{F}(\mathbf{x})\}$.

The region of points defined by the optimal Pareto set in the objective function space is known as *Pareto front*, formally defined as $PF^* = \{(f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})), \mathbf{x} \in P^*\}$.

In a MOP there is not a single optimal solution, but a whole set of non-dominated trade-off solutions instead. Because of this, it is very important to apply specific algorithms for MOP solving, which are able to find many solutions in a single execution. The computed solutions should have good quality and be well distributed to sample the Pareto front of the problem. After that, a decision making procedure is needed in order to select which solution(s) will be applied in practice. This decision making is rarely automatized, and it is often performed by a human.

1.3.4 Multiobjective evolutionary algorithms

Rosenberg suggested the capability of EAs for solving MOPs in a pioneering work back in 1967. The first MOEA was presented by Schaffer in 1984. Since 1990, many MOEAs have been proposed by a growing research community that works actively nowadays. These MOEAs have allowed obtaining accurate results when solving difficult real-life optimization problems in many research areas [10, 11].

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 in each generation. MOEAs must be designed taking into account two goals at the same time: *i*) to approximate the Pareto front and *ii*) to maintain diversity instead of converging to a reduced section of the Pareto front, providing a set of different solutions that represent different trade-offs between the problem objectives. A Pareto-based evolutionary search leads to the first goal, while the second one is accomplished by using specific techniques also used in multimodal function optimization (niches, sharing, crowding, etc.). Figure 1.4 graphically presents these two generic goals for a hypothetical problem that proposes to minimize functions $f_1(x)$ and $f_2(x)$.

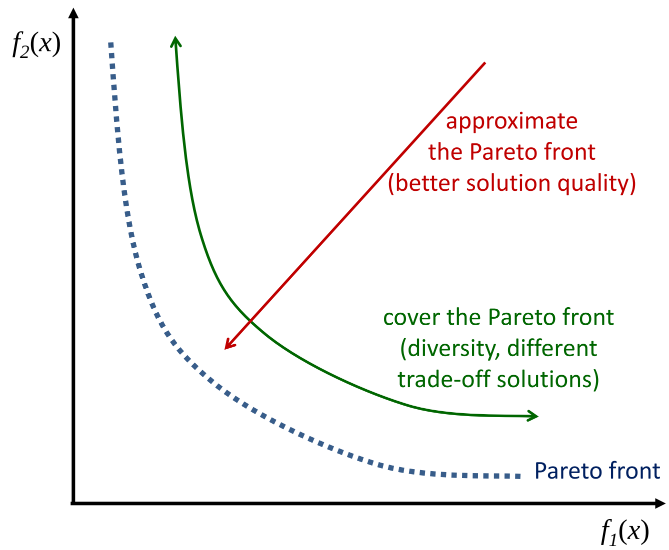


FIGURE 1.4: Goals when designing a MOEA.

In this article, we apply three variants of well-known MOEAs:

- *weighted-sum GA using multiple weight combinations* (WSGA).

  WSGA is a traditional evolutionary algorithm that applies a linear aggregation approach to define the fitness function used to evaluate solutions of the problem. This approach is proposed to allow the evolutionary search to focus on different trade-off section of the Pareto front of the problem, by using several weight combinations.

  In a problem with two objective functions f_1 and f_2, the fitness function in WSGA is defined by the weighted sum of the objective functions of the problem, using a uniformly distributed two-dimensional grid for defining the different weight combinations: $F_{i,j}(x) = w_1^i \times f_1(x) + w_2^j \times f_2(x)$, where $i \in \{1 \dots M\}, j \in \{1 \dots N\}$.

  Although the linear aggregation approach can fail to deal with complex multiobjective problems, in practice it is fully applicable to solve a wide range of optimization problems, especially those having convex Pareto fronts [10]. By sampling the search space using different weight combinations, the linear aggregation approach used in WSGA allows sampling the Pareto front of the problem even though the algorithm does not use an explicit fitness assignment schema based on Pareto dominance.

  Different variants of this aggregation approach have been successfully applied by the authors in the parallel micro CHC algorithm [28] and also to the classic EA in the parallel Multiobjective Evolutionary Algorithm with Domain Decomposition [23, 29].

- *Non-dominated Sorting Genetic Algorithm, version II* (NSGA-II) [12]. NSGA-II is a popular state-of-the-art MOEA that has been successfully applied to solve optimization problems in many application areas. NSGA-II includes features to deal with three criticized issues on its predecessor NSGA: *i*) an improved non-dominated elitist ordering that diminishes the complexity of the dominance check; *ii*) a crowding technique for diversity preservation; and *iii*) a new fitness assignment method that considers the crowding distance values.

  A schema of NSGA-II working on a population P (with size N) is presented in Algorithm 2. The fitness calculation is based on Pareto dominance, building *fronts* of solutions, and crowding distance values to evaluate the diversity and the covering of the Pareto front.

Algorithm 2 Schema of the NSGA-II algorithm.

$t \leftarrow 0$;
offspring $\leftarrow \emptyset$
initialize($P(0)$)
while not stopcriterion **do**
  evaluate($P(t)$)
  R $\leftarrow P(t) \cup$ offspring
  fronts $\leftarrow$ **non-dominated sorting**(R))
  $P(t{+}1) \leftarrow \emptyset$; $i \leftarrow 1$
  while $|P(t+1)| + |fronts(i)| \leq N$ **do**
    crowding distance (fronts(i))
    $P(t{+}1) \leftarrow P(t{+}1) \cup$ fronts(i)
    $i \leftarrow i{+}1$
  end while
  sorting by distance (fronts(i))
  $P(t{+}1) \leftarrow P(t{+}1) \cup$ fronts(i)[1:(N - $|P(t{+}1)|$)]
  selected $\leftarrow$ **selection**($P(t{+}1)$)
  offspring $\leftarrow$ **variation operators**(selected)
  $t \leftarrow t + 1$
end while
return computed Pareto front

- *Strength Pareto Evolutionary Algorithm, version 2* (SPEA2) [38].

  SPEA2 is a popular state-of-the-art MOEA. It has been successfully applied in many problems in diverse application areas. One of the main distinctive features of SPEA2 is the fitness calculation, which is based on both Pareto dominance and diversity: the algorithm defines the *strength* concept to evaluate how many solutions dominate (and are dominated by) each candidate solution, and a density estimation is also considered for fitness assignment. Elitism is also applied, by using an elite population to store the non-dominated individuals found during the search.

SPEA2 was designed to improve over the main drawbacks of the original SPEA algorithm. The main features of SPEA2 include: *i*) an improved fitness assignment scheme, taking into account for each individual how many individuals it dominates and it is dominated by; *ii*) a nearest neighbor density estimation technique which allows a more precise guidance of the search process; and *iii*) an improved archive truncation method that guarantees the preservation of boundary solutions in the elite population.

A schema of SPEA2 working on a population P (with size N) is presented in Algorithm 3. The elite population is elitePop, having a size of eliteSize. When the elite population is full, a pruning method is applied to remove the most similar individuals to assure that the size of the elite population is always eliteSize.

Algorithm 3 Schema of the SPEA2 algorithm.

$t \leftarrow 0$;
elitePop $\leftarrow \emptyset$
initialize($P(0)$)
while not stopcriterion **do**
  evaluate($P(t)$)
  R $\leftarrow P(t) \cup$ elitePop
  for $s_i \in$ R **do**
    $si_{raw} \leftarrow$ computeRawFitness(s_i,R)
    $si_{density} \leftarrow$ computeDensity(s_i,R)
    $si_{fitness} \leftarrow si_{raw} + si_{density}$
  end for
  elitePop $\leftarrow$ nonDominated(R)
  if size(elitePoP) $>$ eliteSize **then**
    elitePop $\leftarrow$ removeMostSimilar(elitePop)
  end if
  selected $\leftarrow$ **selection**(R)
  offspring $\leftarrow$ **variation operators**(selected)
  $t \leftarrow t + 1$
end while
return computed Pareto front

1.4 RSU deployment for VANETs

As presented in Section 1.2, including a RSU platform is crucial for the general deployment of VANETs. However, designing 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 the service provided and network capabilities, while satisfying and/or minimizing the deployment cost constraints. This section presents the RSU-DP formulation proposed and solved in this study. After that, the most relevant studies in the literature about addressing RSU design are reviewed.

1.4.1 The RSU Deployment Problem

The mathematical formulation of the RSU-DP addressed in this study contemplates the following elements:

- The idea is to install a set of RSUs that are defined according to a set of RSU types $T = \{t_1, t_2, \ldots, t_k\}$. In this formulation, the RSU type determines the deployment cost and the communication capabilities (in terms of transmission power and the antenna gain). The type of a RSU is given by the function *type*: $R \rightarrow T$. The deployment cost is computed by the function C: $T \rightarrow \mathbb{R}^+$, where $C(t_h)$ indicates the monetary cost of installing a RSU of type t_h in the deployed RSU infrastructure.

- The RSUs can be located at any place along the considered city streets. Therefore, RSU-DP considers A set of *road segments* $S = \{s_1, \ldots, s_n\}$, which are potential locations for placing a set of RSUs $R = \{R_1, \ldots, R_q\}$. 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). The length of a given segment s_i is given by the function *len*: $S \rightarrow \mathbb{R}^+$. As aforementioned, RSUs can be placed at any location within each segment s_i.

- In addition, an estimation of the road traffic density across each segment s_i in terms of the number of vehicles per time period is given by function NV: $S \rightarrow \mathbb{N}^+$, and the average vehicle speed for each segment is returned by using function *sp*: $S \rightarrow \mathbb{R}^+$.

Solutions to the RSU-DP are defined by a set of RSUs placed over the road segments of the city, represented by a set $sol = \{R_1, R_2, \ldots, R_l\}$ where l (size of sol) is the total number of RSUs in solution sol. In this problem, each RSU is installed in specific coordinates within a segment s_i. However, some road segments might not have any RSU installed on them (l≤n). The segments covered by a RSU are given by the function *cov*: $R \rightarrow S$, and the portion of segment s_i covered by RSU R_j is given by the function *cp*: $R \times S \rightarrow [0, 1]$.

The multiobjective version of the problem proposes to obtain a set of locations and the type of RSU to deploy in each location, with the aim of maximizing the *service time* given by the whole RSU infrastructure, while simultaneously minimizing the *total monetary cost* of deployment. 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 two objective functions in conflict with each other: maximize the service time (QoS), given by $f_1(sol)$ (see Equation 1.1), and minimize the cost, computed by $f_2(sol)$ (see Equation 1.2).

$$f_1(sol) = \sum_{R_j}^{R_j \in\ sol} \sum_{s_i \in cov(R_j)} NV(s_i) \times \frac{cp(R_j, s_i) \times len(s_i)}{sp(s_i)} \tag{1.1}$$

$$f_2(sol) = \sum_{R_j}^{R_j \in\ sol} C(type(R_j)) \tag{1.2}$$

1.4.2 Related Work

The RSU-DP is categorized as a *covering location optimization* problem. It aims at finding optimal positions for RSUs to cover the maximum number of VANET nodes, while taking into account several constraints such as deployment cost, nodes mobility or application requirements. Several authors have addressed the RSU-DP as a variant of other well known covering location problem, the RND problem [25].

The RSU-DP is a hard-to-solve (NP-hard) optimization problem when dealing with (large) instances on city-scaled areas, because the number of possible solutions (i.e., sets of RSU types/locations) becomes very large (virtually impractical) [33]. Therefore, for this kind of real world instances, traditional exact methods are not able to find accurate solutions in reasonable computation times. For this reason, some authors have analyzed the use of specific heuristics and metaheuristics to deal with RSU-DP.

A common approach that is followed by many researchers is to address the RSU-DP considering only intersections (road junctions) as the best tentative locations to install RSUs. The motivation behind this idea is that the vehicle density at the intersections is usually higher than at other points in the roads [1, 7, 20, 35]. However, following this approach can lead to missing accurate solutions in those cases where installing a RSU in some place between the intersections is the best option. Following a more comprehensive approach, the problem model we consider in this chapter allows to install RSUs in every part of the roads, not just intersections.

A review of the main related works on applying exact methods, heuristic, and metaheuristics to the RSU-DP and related problems is presented next.

1.4.2.1 Exact methods

Few articles have proposed applying traditional exact approaches to solve the RSU-DP, mainly because the complexity of the problem does not allow to find solutions in reasonable execution times.

In this line of work, Aslam et al. [1] presented a mathematical formulation of the problem, only considering the intersections as possible locations for RSUs. The formulation proposed to minimize the delay time over the routing path for VANET messages, by installing a fixed number of RSUs in a city scenario. The problem model uses information about speed, road traffic density, and the likelihood of hazardous situations for computing the QoS metric (delay time). An exact Binary Integer Programming method was introduced, but the results computed using this traditional approach were outperformed by applying a Balloon Expansion Heuristic algorithm, when solving a problem instance defined in the city of Miami, USA.

Another proposal of solving the RSU-DP using an exact method was by Liang et al. [19]. However, the proposed resolution method is not able to deal with city-scaled areas, thus simulation results are presented to evaluate the approach on a University campus map (with a total area of less than 3 km^2) and using randomly simulated traffic data. The problem proposes placing RSUs and selecting their configurations (including antenna types, power level, and wired/wireless network connectivity) to minimize the total cost of deployment and maintenance, subject to user specified constraints on the minimum coverage provided by the RSU infrastructure. A scalability analysis is performed over an area with the size of Cambridge, Massachusetts (about 16 km^2), but no comparison against results computed using other methods is presented.

In a recent proposal, Balouchzahi et al. [3] also addressed the RSU-DP using a binary integer programming formulation oriented to minimize the costs of the RSU deployments. A traditional single-objective approach was applied, and both coverage and QoS considerations are included in the form of optimization constraints. The authors observed that vehicles density has a great impact on the network connectivity and this fact is very important for RSU placement. As in many other problem formulations, the candidate locations for placing RSUs are only the intersections between roads in the studied scenario. The proposed method is implemented in Maple and evaluated over urban scenarios (Erlangen, Zurich, Rome, and New York). Each map is divided in zones (200 m$\times$ 200 m each) to allow the approach a proper scalability behavior to solve large problem instances. However, when using this decomposition approach, the solutions depend on the strategy applied to perform the segmentation. The reported results indicate that the proposed algorithm is able to find better solutions in terms of cost, service advertisement time, and service discovery time, when compared against greedy heuristics from the related literature.

1.4.2.2 Heuristics

One of the first works that explored heuristics to solve a RSU-DP-like problem was the one by Trullols et al. [35]. The authors modeled the RSU-DP as the Maximum Coverage with Time Threshold Problem (MCTTP), to maximize the number of vehicles covered by a given number of RSUs for a given predefined period of time. Three greedy algorithms were proposed, based on performing the search by using information about the road topology and the vehicles. The experimental evaluation was performed over an instance built using real data from Zurich, Switzerland. According to the results, the main finding is that vehicular mobility is the most important factor to take into account when deploying an efficient roadside platform. By taking into account the vehicular mobility information, the proposed greedy heuristics allowed to successfully design a VANET infrastructure capable of informing more than 95% of vehicles in the scenario.

In the article by Xiong et al. [37], the RSU-DP is formulated as a Maximum Coverage Problem (MCP). In the proposed problem model, a given geographical area is studied by applying a division strategy, which considers a grid of non-overlapped zones. Over each zone, traffic data is analyzed over a period of several hours, in order to estimate the number of vehicles entering and leaving each zone for each time unit. This procedure allows generating a mobility directed graph that represents the transition probability for vehicles between two zones in the map. Using the information in the graph, the Minimum Gateway Deployment Problem (MGDP) is formulated to find the candidate zones to deploy RSUs, to be used as gateways to provide Internet access. A specific heuristic algorithm named *MobGDeploy* is introduced to find accurate locations for RSUs/gateways. Although the proposed method tries to cover the maximum number of vehicles by choosing to locate gateways in those zones with high vehicle density, it does not take into account any QoS considerations.

A cost-efficient model for RSU deployment to guarantee certain QoS (delay and success ratio) in a file downloading service was presented by Liu et al. [20]. After applying a theoretical analysis to demonstrate the relationship between the RSU deployment strategy and the file downloading QoS, the paper formulates the low cost RSU deployment problem as an optimization problem. The road network is modeled as a weighted undirected graph where each edge is the average passing time on the correspondent road. An heuristic ad hoc algorithm based on the depth-first traversal algorithm for edges of a graph is proposed to solve the problem. The simulation results over a generated road network of 4x4 intersections showed that the proposed method can deploy the RSUs along roads with the lowest cost for this size-limited scenario, meanwhile satisfying the user required file-downloading QoS.

An heuristic approach using a Voronoi diagram-based algorithm was applied by Patil and Gokhale [31] to minimize the number of RSUs to deploy in a vehicular network. In the problem model, the main goal is to optimize the network capabilities in terms of packet loss, communications delays, and network coverage, using the lowest possible number of RSUs. The proposed algorithm evaluates information about the speed of vehicles and the traffic density to compute the quality solutions. However, only one type of RSU is considered. The experimental analysis was performed over a medium-size area (about 32 km^2) in the city of Nashville, USA and using simulated traffic data.

The reported results showed that the Voronoi diagram-based placement algorithm was able to compute solutions with less packet delay and less packet loss when comparing against two simple placement methods: *i*) placing RSUs evenly spread across the area and *ii*) placing RSUs at the busiest intersections). However, the article does not present a comparison against results computed using more powerful placement strategies.

In a more recent article, Ben Brahim et al. [4] solved another variant of the RSU-DP, taking into account real road traffic and mobility data from the city of Doha, in Qatar. Specific pieces of information gathered from the Qatari government were used to define a graph with weighted links to model the problem scenario. In addition to the traditional mobility information (average speed, road traffic density, etc.), Ben Brahim et al. included two other important pieces of information in the problem model to compute the weights of the graphs: the number of hazardous situations and the specific interests on some notable places in the city. A subset of given points in the graph were considered as the potential locations for the RSUs. The problem was addressed by applying two different heuristic approaches: a dynamic algorithm based on 0-1 Knapsack problem solver (*KP_DynAlg*) and the *PageRank* algorithm. The reported results indicate that the PageRank method outperformed the KP_DynAlg algorithm when facing the problem with a given limited number of RSUs. However, in the general case, when the allowed economical budget increased, KP_DynAlg improved the results computed by the PageRank method.

1.4.2.3 Metaheuristics and evolutionary computation

Several studies in the literature have analyzed the application of evolutionary metaheuristic algorithms for solving the RSU-DP. The main goal when applying these approaches is to compute accurate and efficient solutions, often improving over those computed using simple heuristic methods, in reasonable execution times.

An early approach applying a GA to solve the RSU-DP was presented by Lochert at al. [21]. The objective of the proposed optimization problem was to maximize the VANET QoS in a highly partitioned network. A VANET simulator was used to evaluate the tentative solutions (i.e., configurations for locating RSUs). A single problem instance was used to evaluate the proposed approach. The scenario was defined by using real data from the city of Brunswick in Germany, including about 500 km of roads and 10,000 vehicles. The authors did not compare their approach against any other methodology for solving the problem. Instead, they assessed the resulting traffic information system and the optimization strategy by means of simulation.

Later, Cavalcante et al. [7] addressed the MCTTP by applying a GA that uses a greedy method to initialize the population. The proposed GA was evaluated over four different instances defined by using real data from cities in Switzerland. The GA results were compared against those computed applying the heuristic methods previously proposed by Trullols et al. [35]. As the main conclusion, The results Cavalcante et al. showed that the GA solutions guarantee a better vehicle coverage, i.e. up to 11% better than those computed by the greedy approaches.

Another GA to solve the RSU-DP was studied by Cheng et al. [9]. In the proposed GA, the fitness function evaluation took into account the ratio between the road area covered by a given solution and the whole road area. In order to simplify and speed up the computations, the authors used a square grid of $1m \times 1m$. The GA was evaluated over a problem instance defined in Yukon Territory, Canada, by considering geometry-based coverage information about the roads (without including data related to the mobility of vehicles). The evolutionary approach outperformed the α*–coverage* algorithm, a simple heuristic method that proposes placing the RSUs in the center of the road segments.

Our previous works [22, 24] presented an innovative strategy to address RSU-DP: our papers are the first studies that applied an explicit multiobjective approach in order to consider two different conflicting objectives (the QoS of the VANET and the cost) *to be optimized simultaneously*. Specifically, our approach takes into account two relevant problem objectives: *i*) maximizing the coverage in terms of the time that vehicles are connected to the RSUs, and *ii*) minimizing the monetary cost related to the RSU deployment. Our studies consider real information concerning both traffic (speed, traffic density, and road map) from the city of Málaga, in Spain. Real information about the RSUs hardware (costs and network capabilities) is also included to build a realistic city-scaled scenario. Several problem instances were addressed by using a specific MOEA that includes ad-hoc operators. The proposed evolutionary approach significantly improved the results computed using ad hoc greedy methods from the literature.

In the study presented in this chapter, we extend our previous conference paper [22] by considering a set of MOEAs that represent a heterogeneous set of different multiobjective solvers, in order to evaluate the capabilities of each evolutionary method to solve the RSU-DP.

1.5 Multiobjective Evolutionary Algorithms for the RSU-DP

This section describes the features of the proposed MOEAs for the RSU-DP and the greedy heuristics used as a baseline for the results comparison.

1.5.1 Problem Encoding

In the studied MOEAs, the tentative solutions of RSU-DP are represented as vectors of real numbers. Each vector has a length of n, which is the number of elements in the set of road segments S. Each position on the vector stores the information for the corresponding i segment, i.e., the type and the location of RSU to install (if any). The *RSU type* is represented 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). The proposed position within the road segment to locate the RSU 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 1.5 presents an example of solution encoding for a given problem instance with four segments and three RSUs installed. In this example, the first position of the vector corresponds to segment s_1, where a RSU of type 2 (integer part of *2*.16) is installed at $0.16{\times}len(s_1)$ within segment (fractional part of 2.*16*) $s_1 = (p_1, p_2)$. In the second position, the value 1.50 means that the solution proposes to install a RSU of type 1 at the middle of segment $s_2 = (p_2, p_3)$. Finally, the fourth position of the vector (*0*.33) shows that no RSU is installed within segment $s_4 = (p_1, p_4)$ (the fractional part of the value encoded is irrelevant in this case).

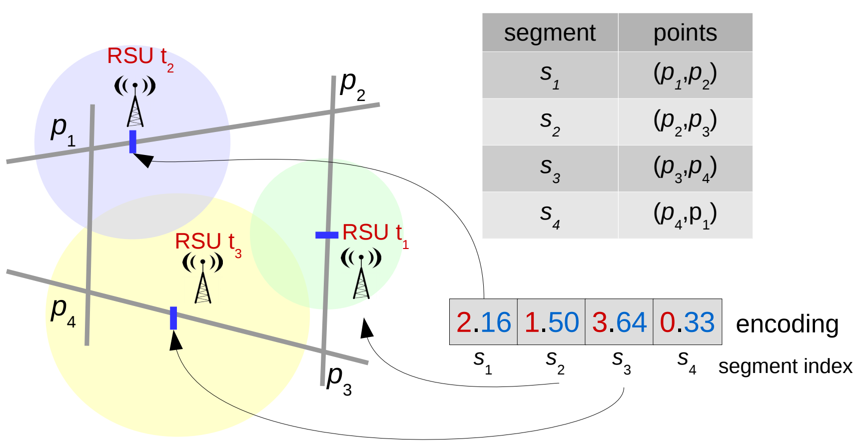


FIGURE 1.5: Proposed encoding for RSU-DP solutions.

1.5.2 Evolutionary Operators

Initialization. In this study, 20% of the solutions of the initial population are seeded by using two randomized greedy heuristics used to compute approximate solutions to the RSU-DP (see a description of the heuristics on Section 1.6.2). This strategy allows the MOEAs to start the evolutionary search from a subspace of good quality solutions. Additionally, the initial population includes the extreme solution of the ideal Pareto front that represents the solution of the minimum monetary deployment cost (i.e., the solution that places no RSU has cost 0). The remaining individuals of this population are initialized by using random real values from the interval $[0,\ k+r]$ being k the number of different RSU types in T, and $r \in [0, 1)$.

Selection. A tournament selection is applied in the three studied MOEAs. The tournament selects two individuals and the best one survives. Applying tournament selection was originally proposed in NSGA-II and SPEA2 [12, 38] to guarantee the preservation of appropriate features of good solutions during the evolutionary search.

Recombination. The *Intermediate Recombination* (IR) operator is applied with a probability p_C in order to recombine genetic information of two solutions or parents and generate two new solutions or offspring. The IR operator is applied over $\overrightarrow{P1} = \{P1_i\}$ and $\overrightarrow{P2} = \{P2_i\}$, which are combined to create offspring $O1$ and $O2$. These new solutions satisfy:

$$O1_i = \alpha_i P1_i + (1 - \alpha_i) P2_i \tag{1.3}$$

$$O2_i = \beta_i P2_i + (1 - \beta_i) P1_i \tag{1.4}$$

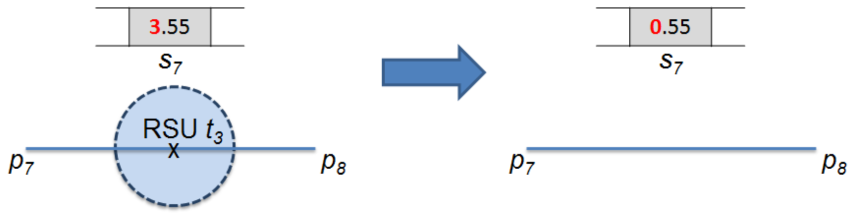
In Equations 1.3 and 1.4, α_i and β_i are real numbers randomly chosen from the interval $[-\mu, 1+\mu]$ for a given value of parameter $\mu \in [0, 1]$.

Mutation. The MOEAs analyzed in this study apply an ad hoc mutation operator specifically designed to address the RSU-DP in order to provide diversity during the search process. This operator is applied with probability p_M. When it is performed, the mutation starts by selecting a number of segments to modify (s_i) according to a uniform probability. Then, it applies one of the following three variations over each s_i segment according to a given probability:

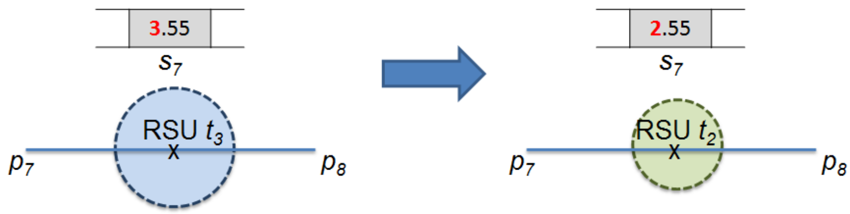
- the mutation operator sets the integer part of the selected gene value to 0, thus removing the RSU (if any) from the corresponding segment (applied with probability π_A, see Figure 1.6.a);

- 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) (applied with probability π_B, see Figure 1.6.b);

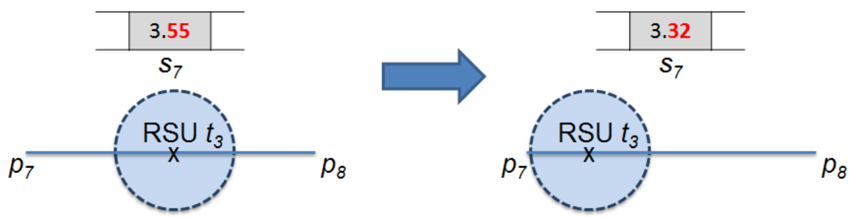
- a *Gaussian mutation* on the value for segment s_i is applied with a standard deviation given by parameter σ in order to change the position of the RSU within the segment (applied with probability $1 - \pi_A - \pi_B$, see Figure 1.6.c).



a) Removing a RSU (applied with probability π_A).



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



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

FIGURE 1.6: Graphical representation of RSU-DP mutation operator.

1.5.3 Evaluation of the Objective Functions

Total cost. The calculation of the total cost is straightforward, by simply adding the cost (according to the corresponding type) of each RSU placed in the scenario.

Quality of Service. For computing the QoS metric (the number of vehicles effectively attended by the RSU infrastructure), we consider the distances and values depicted in Figure 1.7. The RSU placed in segment $s_1 = (p_1, p_2)$ covers the subsegments c_1 (in s_1), c_2 (in s_2) in street A, and c_3 (in s_3), c_4 (in s_4) in street B, according to the coverage defined by the RSU type. The number of effective vehicles attended is given by Equation 1.5.

$$\sum_{i=1}^{i=4} NV(s_i) \times \frac{c_i}{sp(s_i)} \tag{1.5}$$

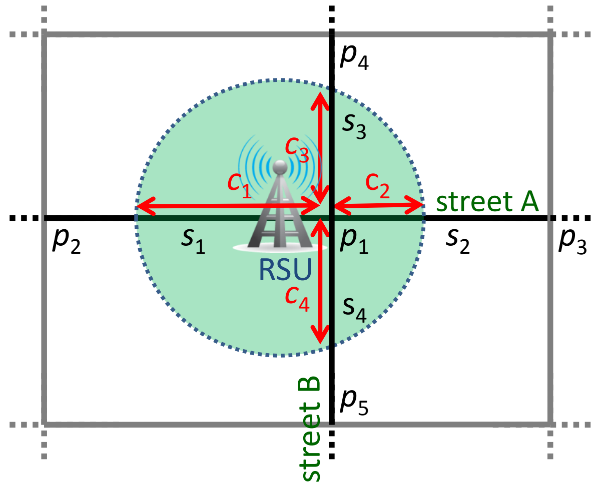


FIGURE 1.7: Calculation of the vehicles attended by a RSU.

The QoS calculation requires computing the intersections between the road segments and the circle representing the coverage of the RSU. We apply a Monte-Carlo simulation approach to compute the length of the subsegment c_i, dividing each segment using 10 equally-spaced points and considering the coverage radius of the corresponding RSU.

Given that the distances involved in the problem are relatively small, we can estimate them using straight lines in the latitude-longitude space with negligible error, instead of using the Great Circle Distance [36]. Thus, we use the Euclidean distance in the latitude-longitude space as an estimation. This approximation makes the QoS evaluation 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 for the corresponding city scenario.

1.6 Experimental Analysis

This section describes the experimental evaluation of the proposed MOEAs to solve the RSU-DP.

1.6.1 Problem Instances

When solving problems such as the RSU-DP, which have real-life applications, it is important to perform the experimental analysis using realistic data and scenarios. In this study, the proposed problem instances are defined by using real world data about the road map, road traffic mobility, and RSUs hardware (network interface/antennae).

The map illustrated in Figure 1.8 shows the area of study in the city of Málaga, which covers 42.557 km^2. Over this area, a set of 128 road segments are defined by 106 geographical points. Some important long roads are represented by more than one unique segment (most notably, avenues like *Avenida de Andalucía*, *Avenida de Valle Inclán*, *Avenida de Velázquez*, and *Paseo Marítimo Pablo Ruiz Picasso*). The road segments have different lengths that vary between 55.5 and 1248.2 m. The average segment length is 483.9 m. All major traffic ways in Malaga, including highways, avenues, and important streets are sampled.

FIGURE 1.8: Map of the area of Málaga and the road segments defined.

The road traffic data is publicly available at the Málaga Council Mobility website [26]. This data was collected by using a set of fixed sensors installed along the roads. These sensors counted the amount of vehicles that circulated between January and June of 2015.

The RSUs hardware is defined by a processing unit equipped with an IEEE 802.11p network interface to perform DSRC communications. Each network interface is connected to an external antenna to extend its communication range and to improve its network capabilities. These antennas are principally characterized by their gain measured in decibels (dBi), which indicates the power of the radio signal radiating from the antenna. Therefore, 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.

In our study, three different types of commercial IEEE 802.11p antennas are considered, according to those available at Cetacea on-line shop [8]. Table 1.1 summarizes the main features of these antennas. Thus, the different RSU types considered vary in their communication range and their price.

TABLE 1.1: Antennas used to define the different RSU types.

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 $

In order to evaluate the performance of a given RSU type, we have evaluated its *effective radio range* (ERR). The EER measures the farthest distance at which the RSU may exchange data packets with the vehicles, while guaranteeing a given QoS. The evaluation of the ERR of each RSU type has been carried out by performing realistic VANET simulations, in which the packet delivery ratio (PDR) of a number of V2R communications was evaluated at different distances (from 0 to 650 m). The experiments were performed using the ns-2 simulator [30] to simulate vehicular communications using IEEE 802.11p PHY/MAC standard in an urban scenario defined by a one lane road of 1 Km using one RSU and 10 moving cars at 40 Km/h. During these simulations, the RSU exchanged data streams at 256 Kbps with the vehicles. In order to obtain realistic results, the non-deterministic fading Nakagami radio propagation model [34] was used to represent the channel characteristics of urban scenarios. The different variants of this VANET scenario were simulated 15 times to compute robust average PDR values. The fourth column of Table 1.1 (*ERR*) summarizes the experimental results by showing the ERR for each RSU type.

1.6.2 Comparison Against Greedy Algorithms

As a baseline to compare the results achieved by the proposed MOEAs, two randomized greedy heuristics are proposed, focused on each of the problem objectives. The greedy algorithms apply intuitive ideas which emulate human-planning strategies for RSU deployment. They are improved versions of the methods defined by Trullols et al. [35] and later used in the comparative study by Cavalcante et al. [7]. The main differences between our heuristics and those presented in [35]are: *i*) in our methods, RSUs can be located anywhere within road segments (instead of allowing RSUs to be placed only at road intersections), *ii*) a variable number of RSUs is considered, instead of using a fixed number of RSUs; and *iii*) a set of RSU types with different costs and coverage are considered, instead of a single RSU type.

The two greedy heuristics developed are described next:

1. *Greedy QoS*. First, the heuristic sorts the set of segments P according to the QoS they provide in case they are totally covered by a RSU (i.e., the ratio between number of vehicles and average speed). The algorithm iterates over the set of sorted segments, processing them in order (segments with better QoS values are processed first), and selects a random position in the segment as a possible location to install a new RSU. Then, two alternatives are considered: *i*) adding to the solution the RSUs that provides the best QoS, in case the QoS can be improved; if two or more RSU types provide the same improvement in QoS, then the cheapest RSU type is selected to be added; or *ii*) not adding a RSU if the QoS cannot be further improved. Segments that are already covered are not taken into account when adding new RSUs.

2. *Greedy cost*. The heuristic starts from the solution computed by Greedy QoS, and then it tries to reduce the cost without significantly affecting the quality of service provided by the solution. Different RSU configurations are explored by replacing existing RSUs by cheaper ones or by deleting RSUs, and the option that accounts for the lower QoS degradation is selected. The algorithm stops when all segments are considered or when the QoS of the solution is equal to $\alpha \times Q$, where Q is the best QoS value achieved by the greedy algorithm for QoS and $\alpha \in [0, 1]$. For the experimental analysis, this algorithm was executed using the values $\alpha \in \{0.70, 0.75, 0.80\}$.

The solutions computed by both greedy heuristics tend to group in different regions of the solution space, depending on the parameters used for their execution. Therefore, in order to evaluate the results obtained by the proposed MOEAs, we compare them against the average results for each one of the four identified groups of greedy solutions, which are labeled *gr_1* through *gr_4* in the figures and tables in Section 1.6.4.

1.6.3 MOEAs Parameter Settings

A parameter analysis was performed in order to find the best values for the parameters in the proposed MOEAs. In the parameter setting experiments, the best results were obtained using the following 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 was set to 0.25. In the Gaussian mutation, the value of parameter σ was set to 0.25. The size of the elite population in SPEA2 was set to 36 individuals. For the weights to use in WSGA, we considered the following sets: $w_{QoS} \in WQ = \{0.0, 0.25, 0.5, 0.75, 1.0\}$, and $w_{cost} \in WC = \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Then, the execution of WSGA was performed using all pairs of weights in the Cartesian product $WQ \times WC$.

1.6.4 Numerical Results

The experimental analysis was performed to assess the problem solving capabilities of the proposed MOEAs for the RSU-DP. We compared the solutions computed by each MOEA against each other and against those computed by the greedy heuristics.

Furthermore, a set of standard metrics for multiobjective optimization [11] were applied in the comparison: *generational distance* (GD), to evaluate the convergence of the computed fronts; *spacing* and *spread*, to evaluate the distributions of solutions; and *relative hypervolume* (RHV) which combines both convergence and dispersion. We also analyzed the Pareto fronts computed by each MOEA in the experimental evaluation over the 20 independent runs that were executed for each of them and for both greedy algorithms.

According to the results obtained in the experimental analysis, NSGA-II showed the best problem solving capabilities among the studied MOEAs. As the reported numerical results demonstrate, NSGA-II significantly outperformed the other two MOEAs as well as the two greedy heuristics, computing accurate Pareto fronts for the problem.

Figure 1.9 shows the global Pareto fronts achieved by the MOEAs as well as the results obtained by the greedy heuristics. Additionally, Figures 1.10 and 1.11 show in more detail the portions of the front were SPEA2 and WSGA achieved its best results, respectively.

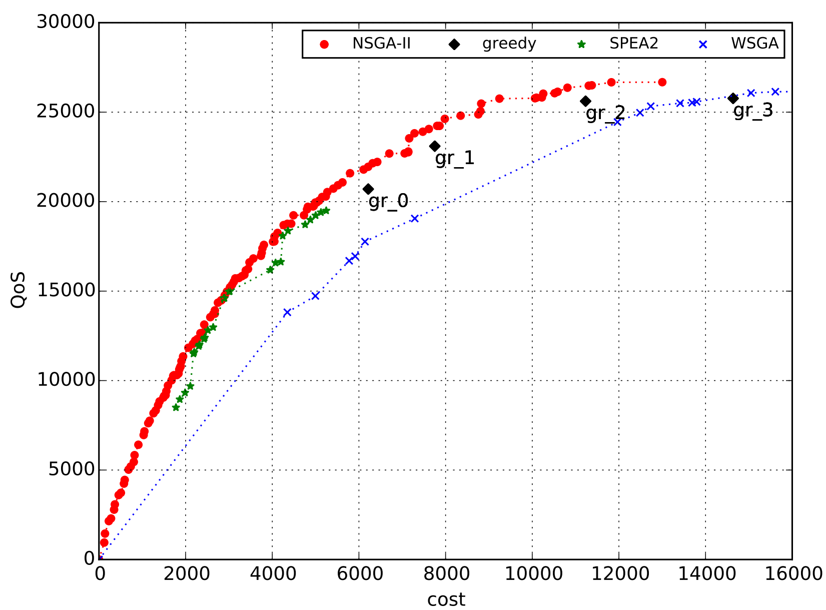


FIGURE 1.9: Global Pareto front and heuristics results

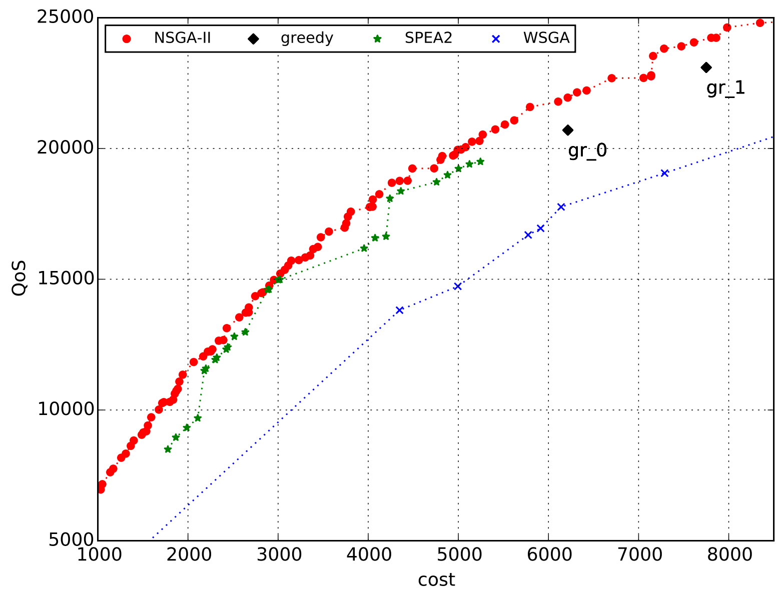


FIGURE 1.10: Global Pareto front and heuristics results (low cost solutions)

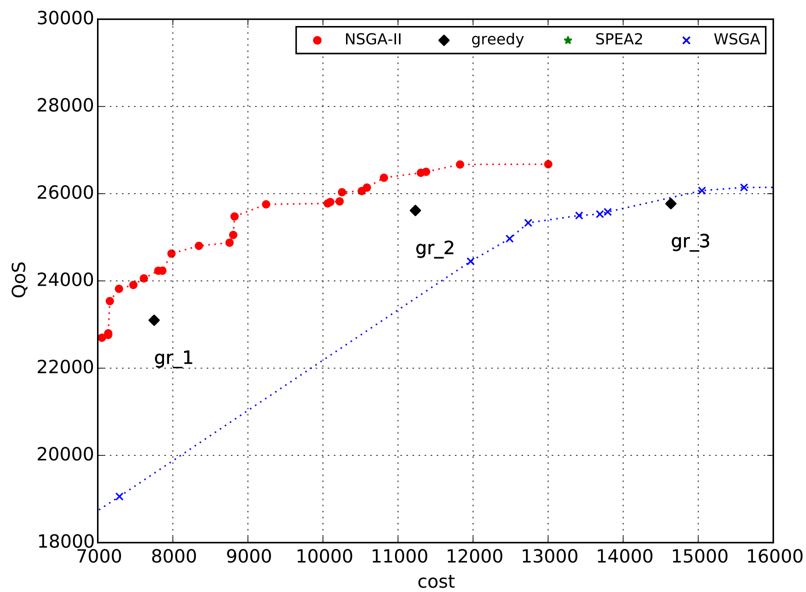


FIGURE 1.11: Global Pareto front and heuristics results (high QoS solutions)

In Figures 1.9-1.11, it can be observed that NSGA-II is able to find a set of solutions that dominates the solutions found by the other two MOEAs and also the solutions computed by the greedy heuristics. Furthermore, the Pareto front computed by NSGA-II is better both in convergence and diversity that the ones computed by the other two GAs. SPEA2 and WSGA tend to converge towards different sections of the Pareto front. SPEA2 finds solutions with low costs (and consequently low QoS) while WSGA concentrates on solutions that are more expensive and with greater QoS.

When using MOEAs, the diversity preservation methods play a major role in finding a good set of non-dominated solutions. The $\mu + \lambda$ strategy applied in NSGA–II seems to maintain a good diversity of individuals throughout the generations, while the elite population in SPEA2 fail to do so, concentrating on smaller regions of the solution space than the region sampled by NSGA–II. The worst results were achieved by WSGA, which indicates that a weighted sum approach to the problem fails to adequately explore the solution space and that Pareto-dominance-based algorithms are a better choice when dealing with the RSU-DP.

NSGA-II is able to improve the QoS of the greedy heuristics in up to **6.0%** while keeping the same cost, and improve up to **36.9%** the cost of the greedy heuristics while keeping the same QoS. Regarding the cost objective, NSGA-II improves over the greedy results 20.3% in average. Improvements on QoS are smaller but still significant: 4.2% in average.

To evaluate the improvements over the greedy algorithms considering the trade-off solutions found for all MOEAs, we take into account the compromise solution (i.e., the solution closest to the ideal vector [11]) computed by each solver. In multiobjective optimization problem, a human decision-maker has the final ruling on which of the non-dominated solutions is selected (in our case, which set of RSUs is deployed). The non-dominated set provides different level of compromise between each objective. The compromise solution offers the best balance between both objective functions, so it is of particular interest for the decision-maker.

Table 1.2 reports the improvements of each MOEA over the greedy solutions (as labeled in Figure 1.9). The compromise solution was computed over the global Pareto fronts achieved by each MOEA, but only considering those solutions with QoS higher than 8000. Solutions with lower QoS (with very low deployment costs, but not providing useful QoS) are not interesting to explore in real life applications.

TABLE 1.2: Improvements of all MOEAs over the greedy strategies.

	cost improvement				*QoS improvement*			
	gr_1	*gr_2*	*gr_3*	*gr_4*	*gr_1*	*gr_2*	*gr_3*	*gr_4*
NSGA–II	-17.2%	6.0%	35.2%	50.2%	15.1%	3.1%	-7.0%	-7.6%
SPEA2	29.8%	43.7%	61.2%	70.2%	-11.3%	-20.5%	-28.3%	-28.7%
WSGA	-92.6%	-54.4%	-6.6%	18.2%	18.1%	5.8%	-4.6%	-5.1%

The results presented in Table 1.2 further emphasize the fact that SPEA2 focuses on solutions having a low deployment cost for the RSU infrastructure, WSGA focuses on solutions with high QoS, and NSGA-II is able to maintain a good balance between both objectives.

Table 1.3 shows the average, standard deviation and best results for the studied standard multiobjective optimization metrics achieved by each MOEA. The ideal Pareto front (which is unknown for the problem instance studied) is approximated by combining the non-dominated solutions obtained over all the executions performed of all the MOEA, as it is commonly done in this research area [10].

TABLE 1.3: Comparison of multiobjective optimization metrics for the studied MOEAs.

	NSGA–II	SPEA2	WSGA
GD	1.5±0.2 (1.2)	10.2±4.7 (0.0)	20.7±0.8 (18.6)
spacing	208.7±21.5 (169.9)	48.3±61.2 (0.0)	8749.8±844.7 (7077.6)
spread	0.4±0.0 (0.3)	1.0±0.0 (0.9)	0.5±0.1 (0.3)
RHV	$1.0{\pm}4.6{\times}10^{-3}$ (1.0)	$0.6{\pm}6.6{\times}10^{-2}$ (0.6)	$0.4{\pm}3.8{\times}10^{-2}$ (0.5)

The good results for the spacing metric achieved by SPEA2 are misleading. Despite the fact that it outperforms the results achieved by NSGA-II, this is due to the small number of non-dominated solutions computed by SPEA2. The spacing metric measures the relative distance between consecutive solutions in the obtained non-dominated set without taking into account the distance to the extreme solutions [11]. Therefore, if the non-dominated set is comprised of few solutions, it is more likely to achieve good spacing values. On the other hand, the spread metric (which takes into account the full Pareto front for the problem, including in the computation the distance to the extremes points of the Pareto front) clearly shows that NSGA–II outperforms both SPEA2 and WSGA. Furthermore, when looking at the RHV, it is clear that NSGA-II outperforms the other two, both in terms of convergence and diversity of the computed solution fronts.

1.7 Conclusions and Future Work

This chapter presents the application of multiobjective evolutionary algorithms to solve the problem of designing an efficient roadside infrastructure to support vehicular networks over realistic urban areas.

The multiobjective formulation proposed in this study considers two conflicting objectives: maximizing the service provided (QoS) by the infrastructure and the monetary deployment cost.

A number of MOEAs were applied to address this optimization problem, including WSGA, NSGA-II, and SPEA2. These algorithms include problem-related operators (encoding and ad hoc mutation) to improve their performance in exploring the search space.

The problem instance used to analyze the proposed methodology was defined on a city-scaled area by using real data (road map, traffic density, and average speeds) from the city of Málaga (Spain). In addition, they take into account three types of real commercial antennas.

As a baseline for our experiments, two state-of-the-art greedy algorithms were developed two address the same problem. Each greedy algorithm solves a single objective: one focuses on optimizing the cost and the other one the QoS. They apply intuitive ideas that emulate planning strategies of a human decision maker in order to construct a solution to the RSU-DP. Among the 3 MOEAs proposed, NSGA-II was the one that achieved better results. In particular, NSGA-II was able to improve the QoS of the greedy heuristics in up to **6.0%** and the cost in up to **36.9%**. Additionally, the solutions computed by NSGA-II converged towards an ideal Pareto front of the problem while keeping better diversity than the one achieved by both SPEA2 and WSGA.

The main future research lines are related to extend the problem formulation in order to include additional relevant road traffic information, such as locations of car accidents and traffic jams or points of interest in the city (schools, industrial areas, etc.). In addition, we plan to take into account different types of VANET applications in order to model more realistic QoS evaluation. Finally, we are working on building a larger set of real RSU-DP scenarios based on real information and areas from different cities, e.g., Montevideo in Uruguay and Cardiff, in Wales, UK.

Bibliography

[1] 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.

[2] T. Bäck, D. Fogel, and Z. Michalewicz, editors. *Handbook of evolutionary computation*. Oxford Univ. Press, 1997.

[3] N. Balouchzahi, M. Fathy, and A. Akbari. Optimal road side units placement model based on binary integer programming for efficint traffic information advertisement and discovery in vehicular environment. *IET Intelligent Transport Systems*, 9:851–861, 2015.

[4] M. B. Brahim, W. Drira, and F. Filali. Roadside units placement within city-scaled area in vehicular ad-hoc networks. In *3^{rd} International Conference on Connected Vehicles and Expo*, pages 1–7. IEEE, 2014.

[5] C2C-CC. CAR 2 CAR Communication Consortium Manifesto. Technical report, CAR 2 CAR Communication Consortium, 2007. [online], available in `https://www.car-2-car.org/index.php?id=31`.

[6] C. Campolo, A. Molinaro, and R. Scopigno, editors. *Vehicular ad hoc Networks - Standards, Solutions, and Research*. Springer, 2015.

[7] E. Cavalcante, A. Aquino, G. Pappa, and A. Loureiro. Roadside unit deployment for information dissemination in a VANET: An evolutionary approach. In *14^{th} Genetic and Evolutionary Computation Conference*, pages 27–34, 2012.

[8] Cetacea. Cetacea Wireless Solutions Company shop. Online at https://shop.cetacea.com/. Retrieved December 2015.

[9] H. Cheng, X. Fei, A. Boukerche, A. Mammeri, and M. Almulla. A geometry-based coverage strategy over urban VANETs. In *10^{th} 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] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. J. Wiley & Sons, Chichester, 2001.

[12] 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.

[13] FCC. Federal Communications Commission news. [online], available in `http://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html`, 1999. Retrieved January 2016.

[14] FCC. Connected Vehicle Research in the United States. [online], available in `http://www.its.dot.gov/connected_vehicle/connected_vehicle_research.htm`, 2015. Retrieved January 2016.

[15] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.

[16] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[17] H. Hartenstein and K. Laberteaux. *VANET Vehicular Applications and Inter-Networking Technologies*. Intelligent Transport Systems. John Wiley & Sons, Upper Saddle River, NJ, USA, December 2009.

[18] O. Johansson, D. Pearce, and D. Maddison. *Blueprint 5: True Costs of Road Transport*, volume 5. Routledge, 2014.

[19] 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.

[20] Y. Liu, J. Ma, J. Niu, Y. Zhang, and W. Wang. Roadside units deployment for content downloading in vehicular networks. In *IEEE International Conference on Communications*, pages 6365–6370, June 2013.

[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 Int. Workshop on Vehicular Inter-Networking*, pages 58–65, New York, NY, USA, 2008. ACM.

[22] R. Massobrio, S. Bertinat, J. Toutouh, S. Nesmachnow, and E. Alba. Smart placement of RSU for vehicular networks using multiobjective evolutionary algorithms. In *2nd Latin America Congress on Computational Intelligence*, pages 1–6, Curitiba, Brazil, October 2015.

[23] R. Massobrio, G. Fagúndez, and S. Nesmachnow. Planificación multiobjetivo de viajes compartidos en taxis utilizando un micro algoritmo evolutivo paralelo. In *X Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, 2015. Text in Spanish.

[24] R. Massobrio, J. Toutouh, and S. Nesmachnow. A multiobjective evolutionary algorithm for infrastructure location in vehicular networks. In 7^{th} *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], available in `http://movilidad.malaga.eu/`. Retrieved December 2015.

[27] S. Nesmachnow. An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*, 3(4):320–347, 2014.

[28] S. Nesmachnow and S. Iturriaga. Multiobjective scheduling on distributed heterogeneous computing and grid environments using a parallel micro-CHC evolutionary algorithm. In *IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2011.

[29] S. Nesmachnow and S. Iturriaga. Multiobjective grid scheduling using a domain decomposition based parallel micro evolutionary algorithm. *International Journal of Grid and Utility Computing*, 4(1):70, 2013.

[30] U. of Southern California. The network simulator ns-2. [online], available in `http://www.isi.edu/nsnam/ns`. Retrieved December 2015.

[31] P. Patil and A. Gokhale. Voronoi-based placement of road-side units to improve dynamic resource management in VANETs. In *Int. Conf. on Collaboration Technologies and Systems*, pages 389–396, May 2013.

[32] M. Picone, S. Busanelli, M. Amoretti, F. Zanichelli, and G. Ferrari. *Advanced Technologies for Intelligent Transportation Systems*. Springer International Publishing, 2015.

[33] 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.

[34] S. Saunders and A. Aragon. *Antennas and Propagation for Wireless Communication Systems*. Wiley, New York, NY, USA, 1999.

[35] 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.

[36] E. Williams. Aviation formulary v1.46. [online], available in `http://williams.best.vwh.net/avform.htm`. Retrieved August 2015.

[37] Y. Xiong, J. Ma, W. Wang, and J. Niu. Optimal roadside gateway deployment for vanets. *Przeglad ElektroTechniczny*, (7):273–276, 2012.

[38] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001.