AC-RDV: A novel ant colony system for roadside units deployment in vehicular ad hoc networks

Abderrahim Guerna[1,2*], Salim Bitam[1], Carlos T. Calafate[3]

1. LESIA Laboratory, Department of Computer Science,
Mohamed Khider University of Biskra, Algeria

2. Department of Computer Science,
Mohamed Boudiaf University of M'sila, Algeria

3. Computer Engineering Department
Universitat Politècnica de València (UPV), Spain

Abstract

Vehicular ad hoc network (VANET) is a mobile and wireless network that consists of connected vehicles, and stationary nodes called roadside units (RSUs) placed on the aboard of roads to improve traffic safety and to ensure drivers' and passengers' comfort. However, deploying RSUs is one of the most important challenges in VANETs due to the involved placement, configuration, and maintenance costs in addition to the network connectivity. This study focuses on the issue of deploying a set of RSUs that is able to maximize network coverage with a reduced cost**.** In this paper, we propose a new formulation of RSUs deployment issue as a maximum intersection coverage problem through a graph-based modeling. Moreover, we propose a new bio-inspired RSU placement system called Ant colony optimization system for RSU deployment in VANET (AC-RDV). AC-RDV is based on the idea of placing RSUs within the more popular road intersections, which are close to popular places like touristic and commercial areas. Since RSU deployment problem is considered as NP-Hard, AC-RDV inspires by the foraging behavior of real ant colonies to discover the minimum number of RSU intersections that ensures the maximum network connectivity. After a set of simulations and comparisons against traditional RSU placement strategies, the results obtained showed the effectiveness of the proposed AC-RDV in terms of number of RSUs placed, the average area coverage, the average connectivity and the overlapping ratio.

[*]Corresponding author.

E-mail addresses: abderrahim.guerna@univ-msila.dz(Abderrahim Guerna), s.bitam@univ-biskra.dz (Salim Bitam), calafate@disca.upv.es (Carlos T. Calafate).

Keywords: VANET; RSU deployment; Intersection-Coverage; Ant colony system; dynamic heuristic function.

1. Introduction

Across the world, and specially in urban areas, every home has typically one or more vehicles; this situation is having a worldwide impact on traffic congestion and road accidents, in addition to having a negative impact on the environment and, in general, on the safety and well-being of citizens. To face this challenge, several efforts have been made to improve traffic management and make transport safer and more comfortable. Therefore, new vehicles are integrated as part of a new system known as an intelligent transportation system (ITS) [1], in which these vehicles operate as nodes of a connected network called vehicular ad hoc network (VANET). VANET is a wireless network based essentially on vehicle-to-vehicle (V2V) Communication mode that ensures message transmission between two or more vehicles being within the same transmission range [2]. The special characteristics of vehicular environments, such as varying driver behavior, high degrees of mobility, and dynamic topology, have an impact on the inter-vehicle link lifetime [3], which is prone to be typically low. To address the challenges associated with V2V communications, including skipping the coverage range limitations of vehicles, Roadside Units (RSUs) can be deployed to provide Vehicle-to-Roadside units communications (V2R). In V2R communications, Roadside Units (RSUs) can play an important role in ameliorating driving safety, traffic management, or even providing drivers and passengers with Internet access [4]. However, despite their many advantages, in the early deployment stages of these technologies, the presence of RSUs is expected to be reduced due to the high deployment and maintenance costs, especially when set on a large-scale. In fact, placing these RSUs, to enhance the vehicular network performance, becomes an important issue, requiring to figure out the optimal places in a given region with a limited number of RSUs in order to achieve maximum network connectivity. In an urban or suburban area, RSUs can usually be deployed at intersections to provide the optimal connectivity performance [5]. In this model, all the intersections were considered as candidate placements. By this way, RSUs placement issue is defined as the process of finding the best combination of RSUs on candidate places according to given conditions to meet the requested requirements (e.g. best connectivity, coverage, low deployment cost). Therefore, the RSU deployment is formulated as a multi-objective optimization problem, with multiple objectives such as maximizing intersection priority (intersection coverage) and minimizing of RSU deployment cost.

This RSU deployment problem is considered as a combinatorial optimization problem [6], and has also been proved to be NP-hard [7]. Unfortunately, for an NP-hard problem, the performance of an exhaustive search is not satisfactory because the number of possible solutions, increasing exponentially with the size of possible solutions (n instances) [27]. Due to its computational complexity, exact algorithms are unsuitable to solve this kind of problem [8]. Indeed, the best solutions of this class of problems are generated using approximate algorithms, often called metaheuristics, leading to quasi-optimal solutions in reasonable computation time.

In this regard, we suggest in this paper a new bio-inspired RSU placement system called: Ant colony optimization system for RSU deployment in VANET (AC-RDV). To the best of our knowledge, the ant colony optimization was not applied in the literature to solve the RSU deployment problem in VANET. AC-RDV is an intersection-coverage algorithm based on the priority concept. Specifically, the AC-RDV Intersection-coverage algorithm aims at placing a limited number of RSUs at intersections to provide the desired connectivity performance. Thus, each RSU placed at any intersection can cover a subset of intersections when these intersections are located within the transmission range of this RSU. Since network coverage is wider at an intersection with dense traffic, compared to an intersection with light traffic [9], we intend to prioritize a subset of intersections to receive the roadside units. Indeed, we consider the idea of intersection priority through the use of the intersection weight concept, as introduced in [10]. The intersection weight is computed using some traffic factors, including vehicles density and intersection popularity. Conceptually, the first RSU can be installed at the intersection with the highest intersection priority, where its coverage includes all intersections within its transmission range, which is referred to as S_i. Thereafter, all intersections belonging to S_i are excluded from the deployment candidate set of intersections, precisely, the updated candidate set has become $\{\backslash S_i\}$. Similarly, the choice of next location for a RSU can be continued until all intersections are covered.

This approach is inspired by the ant colony behavior when searching for food sources, which is discovered within a reasonable time [11]. AC-RDV introduces a new dynamic heuristic function that gives the preference of deploying the $i-th$ RSU at the $j-th$ intersection among the candidates list. When any initial RSU location is determined, we consider an updating network process based on the intersection coverage. To do this, we begin to remove the candidate intersections adjacent to RSU, when these intersections are located within the transmission range of this RSU. The performance of AC-RDV strategy has been evaluated in terms of number of RSUs placed, average area coverage, average connectivity, and the overlapping ratio.

The results obtained showed that the proposed scheme outperformed the traditional RSU placement scheme based on the greedy approach (GA) [10], genetic intersection coverage (GICA) approach previous work [26], and heuristic genetic algorithm (HGA) proposed also in this paper for RSU placement scheme.

The rest of the paper is organized as follows: Related research on the RSU deployment problem is discussed in Section 2. We introduce a transformation of the RSUs deployment problem into the intersection coverage problem, in Section 3. Section 4 presents our proposed optimization schemes. Section 5 details the experimental study and discusses the results obtained. Finally, we conclude the paper and present some directions for future work in section 6.

2. Related work

Several works have been carried out to deal with the RSU deployment problem. Below, we present some relevant approaches.

The uniform distribution of RSUs is the simplest way of deployment in any road network**.** X. Liya [12] presented a randomized algorithm to estimate an approximate optimal distance for deploying RSUs in highways. In their work, a security message can be transmitted to the RSUs from an accident site with a given target probability. However, this approach explores the placement of RSUs that are connected by wired mediums, but this connection strategy is very expensive because of the high number of RSUs placed. Liu et al. [13] analyzed the delay of broadcasting alert messages along a highway. Then, vehicles are grouped into clusters, where cluster members can communicate with each other within no more than two hops. If the vehicle clusters are disconnected, the messages should be carried by vehicles until they encounter an RSU. To obtain the optimal number of RSUs, they derive the relationship between key system parameters such as traffic flow density, transmission range and delay. However, the authors of this study did not propose any RSU deployment strategy.

Extending the logical coverage area of an RSU is another deployment strategy. In [14], the authors developed two optimization methods known as Binary Integer Programming (BIP) and Balloon Expansion Heuristic (BEH) to deploy a small number of RSUs in an urban environment, with the objective of minimizing the reporting average time. However, these proposals consider the network as an ideal graph of nodes and straight lines, which is not the case in the reality. Besides, this proposed method did not analyze the coverage achieved by this technique. The Voronoï diagram is a concept which involves the partitioning of a plane into different convex polygons, where the center point of each polygon (called the generating point) is considered as a favorite location to deploy an RSU. Following this approach, Patil and Gokhale [15] propose a Voronoï diagram-based algorithm to optimize RSU deployment in an urban area. They have used packet loss and packet delay as metrics targeting to minimize these two metrics. However, a placement strategy can involve private land for deployed RSU. Ghorai and I. Banerjee [16] considered that placing the RSUs in an obstructed area is a key concept to achieve full coverage. They suggest a RSUs deployment strategy based on Constrained Delaunay Triangulation (CDT), followed by an optimization procedure to get the best RSUs position and reduce the communication delay in V2R contexts. Thus, an optimal multi-metric RSU selection strategy is introduced. Whereas, the proposed algorithm gives better results in a simple map than in a medium or complex one.

Road intersections with maximum vehicles density are considered as the best potential deployment locations for RSUs. Chi et al [10] presented an RSU deployment approach based on intersection priority approach so that the RSUs are preferably placed at important intersections. The priority of each intersection can be calculated according to some traffic factors including vehicle density, intersection popularity. Greedy, dynamic, and hybrid algorithms are presented to serve this purpose. The greedy algorithm deploys RSUs at intersections in descending order of the intersection priority. The dynamic algorithm concentrates on achieving an even distribution of RSUs in order to reduce the size of the overlapped area. Finally, the hybrid algorithm combines both greedy and dynamic algorithms to distribute RSUs as uniformly as possible, while keeping the order associated to intersection priorities. This approach does not consider the vehicle traffic between intersections to eliminate the overlapping area.

Placing an RSU at an intersection, with dense traffic to minimize the time required for data transmission, is the main contribution of Cavalcante et al [17]. This time is defined as the minimum time required for a vehicle to contact an RSU and successfully transmits information. This problem is solved through a genetic algorithm, and the results obtained are compared to the greedy algorithm. In reality, it is not evident to know the contact time between vehicles and RSUs. However, the model representing communications between vehicles and RSUs is lacked. In [5], the researchers dealt with the RSU deployment problem in an urban area to satisfy the required QoS. They formulate the roadside unit placement problem as a set-coverage problem to provide vehicles with the multi-hop data delivery. Consequently, a Greedy Set-Coverage algorithm is proposed to optimize the number of RSUs and satisfy the required QoS in terms of delivery delay. The results obtained showed that Greedy Set-Cover Algorithm does not always perform well compared to uniform placement. To maximize the number of vehicles that enter an RSU coverage area, the authors of [18] suggest an Integer Linear Programming (ILP) model and a Greedy Randomized Adaptive Search Procedure (GRASP) to optimize the RSUs deployment process. The ILP and GRASP approaches consider the density and the mobility information of vehicles in an urban area. Simulation results showed that this algorithm presents no more than 15% from the optimal value of minimizing the number of roadside units.

Given a limited budget to deploy RSUs, the problem is finding the best locations to install these RSUs so that more roads are covered. Due to the high cost of a massive RSU deployment in wide metropolitan areas, Kim et al. [19] suggests a new strategy to optimize RSU deployment using three different deployment techniques, i.e., static locations, the public transportation units that are not controllable (i.e. Buses) and fully controllable mobile nodes (i.e. vehicles). The simulation results showed that this framework provides a cost-effective solution compared to the case of adopting a single deployment strategy. However, this work considers that each mobile transportation does not suffer from any delay and the controllable mobile does not suffer from traffic jam, which is not the case in a real world scenario.

In [26], the authors formulated the RSUs deployment problem as a multi-objective optimization problem, hence they proposed a new genetic intersection-coverage algorithm (GICA) based on the priority concept. In this work, the purpose is to focus on popular intersections in terms of RSUs installation, aiming to maximize the coverage of RSUs while minimizing the interference rate and RSUs costs. The tests leaded to prove that GICA has better results over greedy approach, but it does not take into account the average connectivity and deployment budget variation.

In summary, most of the works presented above have focused on optimally deploying a limited number of RSUs to improve network coverage, but they did not consider the variations in data traffic, which depend on critical parameters such as placement location, deployment budget, and road topology. Therefore, we propose in this paper a new bio-inspired RSU placement system called Ant colony optimization system for RSU deployment in VANET (AC-RDV), aiming at placing a reduced number of RSUs that cover a large geographic area, and improve network connectivity with a limited overlapping ratio.

3. System model

As for the deployment problem in vehicular networks, [17][18][19] consider the road intersections as the best location to deploy RSUs. In urban road topology, many intersections exist; however, deploying a large number of RSUs is a costly solution. Therefore, the RSU deployment is formulated as a multi-objective optimization problem, which includes maximizing intersection priority (intersection coverage) on one hand, and on the other hand, it minimizes RSU deployment cost. In this section, the problem description and some definitions are discussed, to be used in the rest of this work.

3.1. Problem Description

The first objective of this work is to answer how RSUs can be deployed in urban VANET. Therefore, allocate the RSUs at intersections that have a higher impact on the efficiency of the vehicular networks is the best deployment strategy. The main benefit of this strategy is to deploy the RSUs at high priority intersections in order to maximize the coverage for vehicles within a monitored area.

Definition 1 Urban Road Map: This can be represented as an undirected graph, $G = (I, E)$. $I = \{I_1, I_2, \ldots., I_n\}$ and $|I| = n$, denotes the intersections set that represents candidate sites for placing RSUs. $E = \{E_1, E_2, \ldots., E_m\}$ is segment roads set, and $e_{ij} \in$ E is the road segment connecting two intersections I_i and I_j. Furthermore, $d_{i,j}$ indicates the distance between two RSUs located at I_i and I_j.

In order to maximize the number of vehicles circulating near an intersection, we associate with this urban road map a weight function:

$$P : I \rightarrow \mathbb{R}^{+}$$

$$I_i \mapsto p_i$$

Regardless intersection $I_i \in I$ of graph G, the weight p_i of each intersection represents the importance of each intersection. In other wording, we use the concept of “Intersection Priority”.

Definition 2 Intersection Priority [10]: can be calculated according to m traffic parameters. The priority of the $i - th$ intersection is determined as follows:

$$\boldsymbol{p_i = \sum_{j=1}^{m} w_j . f_{ij}} \qquad (1)$$

Where $\boldsymbol{f_{ij}}$is a normalized value obtained by the j_th traffic factor for the i_th intersection and $\boldsymbol{w_j}$ is a weight for each traffic factor, where $1 \leq j \leq m$. Thus,

$$\boldsymbol{\sum_{j=1}^{m} w_j = 1} \qquad (2)$$

In this work, we use vehicle density and location popularity as traffic parameters. The vehicles density is measured as the total number of vehicles that crosses each intersection for each time unit, while the intersection popularity denotes the geographical importance of each intersection.

We also define that an intersection is covered by an RSU if the intersection is located within the transmission range of the RSU (R).

Recall that our goal is to cover all the road segments of a graph $G = (I, E)$ with a minimum number of RSUs. According to the graph theory and combinatory optimization, this problem can be formulated as a classical optimization problem known as the "minimum vertex coverage problem" [21].

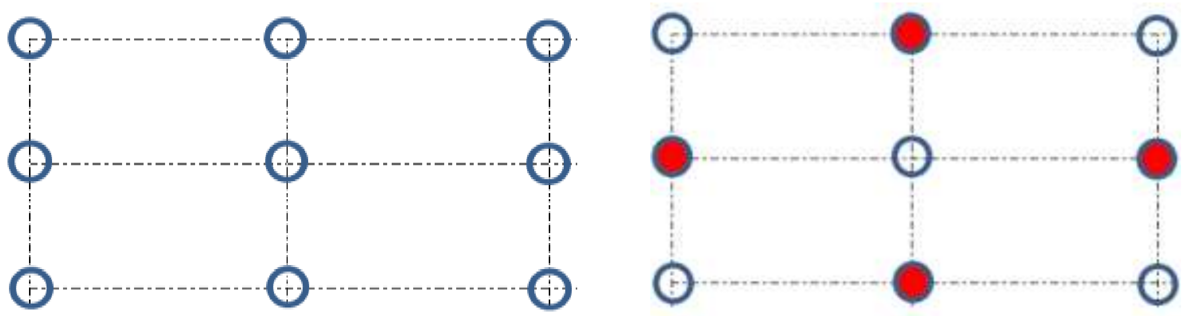
In order to place the RSUs at high priority intersections, we employed two sets indicated as $RSET$ and $CSET$. At the beginning, the RSET subset defines a highly prioritized intersection list that allows determining the location of the first RSU. Thereafter, all intersections within the transmission range of this RSU are excluded from the candidate set of intersections for deployment. Notice that $RSET$ contains all intersections where RSUs are placed, on the other hand, $CSET$ includes all intersections covered by RSUs placed at $RSET$.

Definition 3 Intersection-coverage: An intersection I_j is covered by an RSU placed at an intersection I_i if I_j is located within the transmission range R of this RSU.

So, $\forall\ I_i \in RSET$, $\forall\ I_j \in \{I \setminus RSET\}$

$$(L_{ij} \leq R) \Leftrightarrow (I_j \in CSET)$$

In this case, intersection I_i covers the intersection I_j.



a) Input graph.

b) The intersection in red are the RSET set, while the other intersections build the CSET set.

FIGURE 1 Example of the intersection coverage problem.

As shown in Figure 1, a RSU coverage of a road trace $G = (I, E)$ consists of finding a subset $RSET \subseteq I$ of all road intersections, where $|RSET| \leq K$ is the optimal subset of intersections that are selected for RSU deployment, satisfying the following conditions:

$$\begin{cases} RSET \cap CSET = \emptyset \\ \quad\quad end \\ RSET \cup CSET = I \end{cases}$$

For each intersection $\boldsymbol{I_i}$, we have a decision variable $\boldsymbol{x_i}$if a RSU is placed at the $i-th$ intersection,$\boldsymbol{x_i} = 1$,otherwise $\boldsymbol{x_i} = 0$.

In our model, the vehicles must be connected with neighboring RSUs, and so the goal is to deploy RSUs at high priority intersections aiming to maximize the coverage for vehicles within a monitored area. According to this goal, a linear programming formulation for our problem can be provided as follows:

$$Z = max\left[\frac{\sum_{i\in I}(p_i * x_i)}{\sum_{i\in I} x_i}\right] \quad (\mathbf{3})$$

Subject to:

$$\sum_{I_i\in I} x_i \quad \leq K \quad (\mathbf{4})$$

$$d_{i,j} \quad \geq 2.R\ , \ \ \forall\, i,j \ \in RSET \quad (\mathbf{5})$$

$$x_i \in \{0,1\} \quad \forall\, I_i \in I \quad (\mathbf{6})$$

The objective function (3) favors more the intersections with high priority, while minimizing the number of these intersections. p_i denotes the priority of the $i-th$ intersection. Constraint (4) ensures that the coverage of all the road segments by the RSUs does not exceed a maximum threshold K. In order to avoid overlapping coverage cases, the distance between two neighboring RSUs installed in adjacent intersections i and j will account for the transmission range of the RSUs. To achieve this, we introduce two sets denoted as RSET and CSET. RSET includes all intersections where RSUs are placed, while CSET contains all intersections covered by the RSUs included in RSET. This constraint is defined in (5). Constraint (6) defines the integrality constraints.

3.2.Heuristic Genetic Algorithm (HGA)

In this section, we propose an enhancement of a genetic algorithm presented in our previous work [26]. Called Heuristic Genetic Algorithm (HGA), it has a standard structure contains a set of operations such as, coding and initialization, crossover, mutation, and reproduction. The initial population is generated randomly to ensure more diversity of the solving process. However, this random initialization technique leads to a very slow convergence to the optimal solution. To speed up the research process to the global optimum, a new initial population method has been suggested in this algorithm, named Greedy Heuristic Initialization (GHI), GHI represents an original population initialization that increases the quality of initial population (this initialization is presented in the Algorithm 2). HGA algorithm 1 proposed to solve the RSUs deployment problem is as follows:

Algorithm 1. HGA

Input: $\mathbf{G(I,E), p_i, D, i = \{1,\dots,n\}}$

Output: RSET

1: Initialize parameters $R, p_{Cross}, p_{Mut}, \varrho, \psi$

2: Heuristic initial population $P^{(t)}$. // using algorithm (3)

3 : $P^{(t)}$ where $|P^{(t)}| = Tett = 0$

4 : $best^{(0)} \leftarrow \max\left\{z_f^{(0)}\right\}$, $j = \{1,\dots,T\}$

5: **While** ending condition is not met **do**

6: $P'^{(t)} \longleftarrow$Crossover($P^{(t)}, p_{Cros}$)

7 : $P'^{(t)} \longleftarrow$Mutation($P'^{(t)}, p_{Mut}$)

8: Evaluation ($P^{(t)}, P'^{(t)}, z$) // using formula (3) as fitness function

9: Select ϱ children using the Roulette Wheel Selection

10: Insert $(n - \varrho)$ elitist parents in next population $P^{(i+1)}$

11: $best \leftarrow \max\ \{best, z_{.}^{(t+1)}\}$

12**: End while**

13: **return** best solution $RSET^{best}$

14: **End**

- *Individual coding and initialization*

In our preview study [26], the proposed GICA algorithm encodes solution using a binary array (it is an individual) of n positions, where each bit position stands for one RSU location. For this purpose, a bit is set to one if and only if there is a RSU at this position. For instance, let's consider 4 RSUs and *n=10* a valid solution individual {0, 3, 4, 6, 9}, i.e., the RSUs are placed into intersections: $\{I_0, I_3, I_4, I_6, I_9\}$, i.e., the individual solution is represented by: (1,0,0,1,1,0,1,0,0,1), there are ten possible position of RSUs, five of them are used.

After the coding step, we generate a set of solutions consisted of T individual, known as initial population $P^{(0)}$. Usually, the initial population is created randomly without any rules (prior experience). However, the random initialization would cost very long time to get the near-optimal solution [28]. To solve this problem, we replace the random initialized population by a greedy heuristic initialization (GHI) in order to speed up the generation of good population. Indeed, Greedy Algorithm [10] is modified to generate a set of feasible solutions, where, each feasible solution contains vector of a set of RSU positions. At each iteration, we allow any random intersection among the top $(n/4)$ ranked ones to be chosen (not only the highly priority intersection is selected). By this way, our proposed algorithm could start with better (fitter) individuals as an initial population generated as mentioned by algorithm 2 (presented below).

Algorithm 2. GHI

Input: p_i, $i = \{1, \dots, n\}$

Output: P // Initial population

Sort I with p_i in a descending order of priority

$P \leftarrow 0$;

for j:=0 to T **do**

Select $I_i \in I$ where i = rand (0 : $n/4$)

$P_{Ji} \leftarrow 1$

$I \leftarrow \{I \backslash S_i\}$ // S_i the coverage of RSU located at I_i

$i \leftarrow i + 1$

while I $(! = \emptyset)$ **do**

Choose $I_i \in I$ where p_i the highest is

$P_{Ji} \leftarrow 1$

$I \leftarrow \{I \backslash S_i\}$

$i \leftarrow i + 1$

end while

end for

return P

End.

- *Crossover:* it is a binary operator that recombines the two selected parents to generate two children according to a probability p_{Cros}. We select randomly two random crossover points $cr_1$and cr_2; then; the genes limited by $cr_1$and $cr_2$are swapped between the parent individuals in order to generate two new children individuals.

- *Mutation:* it is a unitary operator that introduces the diversity into the population; i.e., preventing the research process to fall into local minima solutions [23]. This operator acts on an individual, for each gene, meaning that a gene may vary its value using a constant probability p_{Mut}, which is very small, so that it will not develop into an intolerable influence.
- Reproduction*:* According to the fitness function, the individuals with a best fitness will be selected to form the next generation. The reproduction operator works in two steps. First, it uses the "roulette wheel" procedure to select the ϱ children's individual. Second, it applies an "elitism" selection by copying the ψ parent's individual having the highest fitness value. So, the size of new population is $n = \varrho + \psi$.

Stopping criterion: In this algorithm, the main loop is iterated until reaching a fixed number of generation $\boldsymbol{t}$ (see the algorithm 1).

4. Ant colony system for the RSU deployment problem

In this section, we present the proposed Ant Colony System (ACS), which is one of the ACO variants [9]. First, we will provide a brief introduction on the principles underlying the ACS algorithm, and then we will present the details of the AC-RDV to optimize RSU deployment.

4.1. Ant colony system

In an Ant Colony System (ACS) a set of agents (called artificial ants) cooperate in finding good solutions to combinatorial optimization problems. This approach, due to Dorigo [22], is inspired on the collective behavior of ants that communicate with each other indirectly via a chemical substance known as the pheromone, allowing the ants to establish an optimal path between their nest and the food source. The Travelling Salesman Problem (TSP) was the subject of the first implementation of an Ant Colony System [23].In this case, the ACS works as follows: initially, ants are randomly positioned on different cities. With each edge between a pair of neighboring cities(i, j) is associated to pheromone value $\tau_{ij}$and heuristic informationη_{ij}. The heuristic information represents a priori information about the problem instance. Solving TSP means finding the shortest possible route towards each city and returns to the starting city. So, the heuristic information is calculated as the inverse proportional of the distance between two neighboring cities. Each ant finds a solution by moving through a (finite) sequence of neighboring cities. Consequently, these moves are selected according to an iteratively stochastic procedure that combines use of pheromone trails and heuristic information. While finding its solution, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once all ants have terminated their tour (solutions), the amount of pheromone on edges is modified again (by applying the global updating rule). This pheromone information will direct the search of the future ants. In the following we discuss the formulation of AC-RDV algorithm for RSUs deployment problem. It consists of the different stages: state transition rule, the global updating rule and the local updating rule. In the following, we will give details of these steps for the RSUs deployment problem.

4.2. AC-RDV Approach

Since the RSU deployment is a discrete optimization problem [24], the Ant Colony System (ACS) emerges as an efficient approach for solving this kind of problem [25]. Generally, the research process of ACS is composed of two loops that are interrelated. The first one is the research cycle of individual ants, which finishes when the ant happens to cover all the graph edges. The second one consists of combining the individual results of all the ants to make a global solution to the problem (see AC-RDV algorithm). At the beginning of algorithm, m ants are released and randomly choose their starting intersection; then, each of them starts to make a solution to the problem by filling on a list with one intersection at each step until it can cover all the graph edges (road segments).During the research process, an ant l chooses the following intersection by counting the combination of the pheromone trail values and the heuristic information. Then, it privileges the intersection characterized by a higher probabilistic value (see equation7). Every ant will have memory regarding the intersections it has already selected in order to guarantee the validity of the constructed list. Figure 2 represents the ant decision depending on both the pheromone trail τ_j and the heuristic information η_j gathered, where $j \in \{C, D, E\}$.The decision to pick an intersection $\boldsymbol{j}$ when the ant is atintersection $\boldsymbol{i}$ for time step $(t)$is obtained as follows:

$$\boldsymbol{a_{ij}(t)} = \frac{\boldsymbol{\tau_j \cdot (\eta_j)^{\alpha}}}{\sum_{\boldsymbol{k \in A^i}} \boldsymbol{\tau_k \cdot (\eta_{kt})^{\alpha}}} \qquad \textbf{(7)}$$

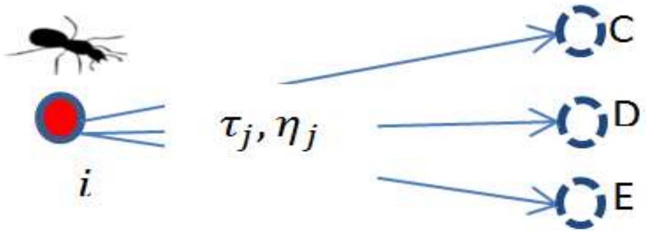


FIGURE 2 Ant decision depending on τ_j and η_j.

Where $A^i$is the set of the intersections that are neighbors to ant $l$when located of intersection I_i. τ_j is the pheromone trail on intersection I_j, η_j is the heuristic information, and $\alpha \geq 0$ is the parameter controlling the relative influence between the heuristic information and the pheromone trail. The heuristic information $\eta_j$determines the local favorableness of choosing the $i-th$ *intersection* that has the best value in terms of intersection priority.

In ACS, a new state transition rule called pseudo-random-proportional is introduced [11]. Depending on the pheromone trail and the heuristic information, the ant l located at an intersection I_i chooses the intersection I_j as its next intersection to be visited according to two parameters: q_0 and q. Let $q_0 \in [0,1]$, which is the parameter specifying the compromise between exploitation of the recent solution and exploration of other unvisited or relatively unexplored search space regions, and q is a random variable uniformly distributed over [0, 1].The pseudo-random-proportional transition rule is given as follows:

$$\boldsymbol{P_{ij}^l(t)} = \begin{cases} \mathbf{1} & \boldsymbol{si}\ q > \boldsymbol{q_0}\ \boldsymbol{and}\ \boldsymbol{j} = \mathbf{Arg\,max}\,(\boldsymbol{a_{ij}})\ \forall\, \boldsymbol{j} \in \boldsymbol{A^i} \\ \mathbf{0} & \boldsymbol{si}\ q > \boldsymbol{q_0}\ \ \boldsymbol{j} \neq \mathbf{Arg\,max}\,(\boldsymbol{a_{ij}})\ \forall\, \boldsymbol{j} \in \boldsymbol{A^i} \\ \dfrac{\boldsymbol{a_{ij}(t)}}{\sum_{\boldsymbol{k} \in \boldsymbol{A^i}} \boldsymbol{a_{ik}(t)}} & \boldsymbol{si}\ \boldsymbol{q} \leq \boldsymbol{q_0} \end{cases} \qquad (\mathbf{8})$$

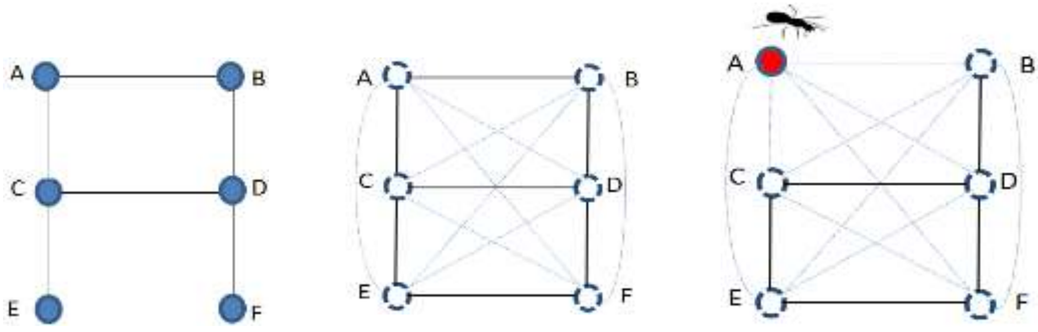
Concerning the performance of AC-RDV algorithm, the heuristic information η_j plays an important role; it takes the objective function into consideration in the process of finding a solution. However, there can be two ways to define heuristic information: static or dynamic [25]. Here, we devise a dynamic heuristic to reflect the reality that the number of road segments that are not yet covered will change whenever an RSU is deployed.

4.2.1. Dynamic heuristics and graph updating

The heuristic function is the ratio between the temporary degrees of an intersection and intersection priority. The temporary degree of an intersection $\boldsymbol{I_j}$ is defined as the number of road segments covered by intersection $\boldsymbol{I_j}$, but not covered by any intersection $\boldsymbol{I_i} \in RSET_{k-1}$, where $RSET_{k-1}$is the partial solution in step $k-1$ (before adding intersection $\boldsymbol{I_j}$ to the solution).

In another wording, an intersection $\boldsymbol{I_i}$ is covered by an RSU placed at intersection $\boldsymbol{I_j}$ if the distance between $\boldsymbol{I_i}$ and $\boldsymbol{I_j}$ is less than or equal to $2R$. Let S_i be the coverage of RSU located at $\boldsymbol{I_i}$ (intersection-coverage of $\boldsymbol{I_i}$), that includes all intersections within the transmission range of this RSU. To model the coverage of an intersection i with another $\boldsymbol{I_j}$, it is natural to use a strongly connected graph $G_c = (I, E_c)$ derived from graph G. So, the temporary degree is given by the decision variable $\gamma_k(\mathrm{i,j})$. When an intersection $\boldsymbol{I_i}$ is covered by an intersection $\boldsymbol{I_j} \in RSET_{k-1}$,

$\gamma_k(\mathrm{i,j}) = 1$; otherwise, $\gamma_k(\mathrm{i,j}) = 0$. Where $(\mathrm{i,j})$ is the link between the two intersections $\boldsymbol{I_i}$ and $\boldsymbol{I_j}$. This G_c graph must be updated once a new intersection $\boldsymbol{I_j}$ is introduced to RSET, i.e. all intersections belong $\boldsymbol{S_i}$ are excluded from the deployment candidate set of intersections; therefore, the temporary degree changes. The graph updating is shown in Figure 3.



a) Original graph G b) Derived graph G_c c) Update graph G_c

FIGURE 3 The coverage updating graph

So, the heuristic function will be dynamically evaluated and calculated as follows:

$$\boldsymbol{\eta_{jk}} = \frac{\sum_{(i,j)\in E_c} \boldsymbol{\gamma_k(i,j)}}{\boldsymbol{p_j}} \qquad \textbf{(9)}$$

Where k is the number of added interactions, $\sum_{(i,j)\in E} \boldsymbol{\gamma}_k(i,j)$ is the temporary degree of intersection $\boldsymbol{I_j}$, and $\boldsymbol{p_j}$ is the priority associated to intersection $\boldsymbol{I_j}$.

The selection of those intersections $RSET \subseteq I$denotes the optimal location for the RSU deployment and the road segments that should be covered.

4.2.2. Pheromone updating

In the AC-RDV algorithm, pheromone updating consists of two rules: local update and global update. The local pheromone update is defined when an ant l at an intersection i chooses a new intersection $\boldsymbol{I_j}$ to its partial solution S^l. Ant l updates the amount of pheromone $\tau_i$according to the following formula:

$$\boldsymbol{\tau_i = (1 - \varphi).\tau_i + \varphi.\tau_0} \qquad (\mathbf{10})$$

Where $0 \leq \varphi \leq 1$ is a parameter used to specify the strength of the local update rule. Once all the ants have made their solutions, the pheromone traces are updated as follows:

$$\boldsymbol{\tau_i(t+1) = (1 - \rho).\tau_i(t) + \rho.\Delta\tau_i} \qquad (\mathbf{11})$$

Where, $\rho \in [0,1[$ is the coefficient that will define the rate of evaporation of the pheromone on the intersection between iterations t and $(t+1)$. Regarding $\boldsymbol{\Delta\tau_i}$, it provides the quality of the best subset I' which contains intersection $\boldsymbol{I_i}$:

$$\boldsymbol{\Delta\tau_i} = \begin{cases} \boldsymbol{1/\sum_{j\in I'} p_j} & \boldsymbol{if} \quad \boldsymbol{i} \in I' \\ \boldsymbol{0} & \boldsymbol{otherwise} \end{cases} \qquad \mathbf{(12)}$$

The stop criteria of our algorithm are the maximum number of iterations.

Algorithm 3: AC-RDV

Input: $G(I,E)$

Output: Neighborhood map of RSUs based on intersection priority p_j;

Initialize parameters $\rho, \tau_{ij}, p_{best}, \tau_0, \varphi$;

Initialize the ants number l ;

Best solution : $RSET = \emptyset$;

while ending condition is not met **do**

Construct a complete graph $G_c = (I, E_c)$

for all ant from : 1 to l **do**

Get the initial graph G

Repeat // Each ant is positioned on an arbitrary starting node

Compute η_j based on (9)

For each ant choose the next intersection using the state transition rule (8)

Update graph $G_c = (I, E_c)$

Apply the local pheromone update rule based on (10)

Until no intersection visited.

End for

Apply the global pheromone update rule according to (11)

Return the solution of each ant ($RSET$ and CSET)

Calculate the overlap area of each ant

End while

return best solution $RSET^{\text{best}}$

End

4.3. Computational Complexity analysis

Usually, the computational complexity of any algorithm is measured in worst-case complexity; it is denoted in asymptotic notation that is indicated the longest running time performed by an algorithm given any input of size $\boldsymbol{n}$. Computing the computational complexity of any algorithm involves the estimation of the number of elementary steps performed to finish execution. According to this proposal, from step 9 to 13, denote the solution cycle, the ants make (in worst case) n visits to build solution. For the l ants, the computational complexity is estimated as $\boldsymbol{O}(\boldsymbol{l}.\boldsymbol{n}^{2})$**.** Since, it is a complete graph, the complexity in step 5 is given by $\boldsymbol{O}(\boldsymbol{n}^{2})$, where $\boldsymbol{n}$ is the graph order (the number of vertics). From step 15 to 17 the complexity is $\boldsymbol{O}(\boldsymbol{2n}+\boldsymbol{l})$. Finally, the computational complexity of one iteration of the proposed AC-RDV algorithm, therefore, it becomes: $\boldsymbol{O}((\boldsymbol{l}+\boldsymbol{1}).\boldsymbol{n}^{2}+\boldsymbol{2n}+\boldsymbol{l}) \approx \boldsymbol{O}(\boldsymbol{n}^{3})$**.** For a maximum number $\boldsymbol{NC_{max}}$of iterations, the general complexity of the algorithm is: $(\boldsymbol{NC_{max}}.\boldsymbol{n}^{3})$ **,** where NC_{max} is a constant belonging to $\mathbb{N}$**.** On the basis of this complexity function, our algorithm can give better near-optimal solutions in polynomial time.

5. Performance evaluation

In this section, we evaluate the performance of the proposed optimization strategy and present the results obtained. We analyze how our algorithm works differently according to the different characteristic of road networks and finds the optimal number and locations of the RSUs deployed in such areas. Therefore, we use three random topologies classes including 67 intersections, 72 intersections and 224 intersections. However, each topology makes a variation of the number of road segments to build three different instance classes of network topologies. To be more realistic, the network topologies have been generated randomly including the positions of intersections. In order to measure the priority of each intersection, two traffic factors are taken into account: regarding (1) the density of vehicles and (2) the intersection popularity. The vehicles density refers to the volume of traffic at each intersection, while the intersection popularity describes the geographical interest of the intersection. Hence, the popularity of an intersection is measured by the different bus lines passing through it. These parameters are obtained randomly with respect to a uniform distribution, either from the interval based on the traffic data provided in [10]. Table 1 details the three network topologies used during the evaluation process in terms of number of roads and intersections parameters.

5.1. Baseline and evaluation metrics

For evaluating the effectiveness of our algorithm, we use four performance metrics: the number of RSUs, the average area coverage ($\boldsymbol{Cov}$) by the RSUs, the average connectivity ($\boldsymbol{c_n}$) and the overlapping ratio (δ). The average area coverage by the RSUs indicates the ratio of road segments coverage in the network. The average connectivity ($\boldsymbol{c_n}$) refers to the ratio of the corresponding intersections $I_i \in CSET$ and the total number of intersections. The overlapping ratio is denoted as: $\boldsymbol{c_n = \frac{1}{n}\sum_{i=1}^{n}\frac{|S_i|}{n}}$. Where $\boldsymbol{S_i}$ denotes the intersections set belonging to transmission range of the RSU installed at $i-the$ intersection. The overlapping area of the $i-th$ RSU is denoted as: $\delta_i = \sum_{\mathrm{I_j \in RSET}} (2.\mathrm{R} - d_{ij})$; where $\mathrm{d_{ij}} < 2.R$. The overlapping ratio is denoted as: $\delta = \frac{1}{n}\sum_{i=1}^{|RSET|}\sum_{j=i+1}^{|RSET|}\frac{(2.\mathrm{R}-d_{ij})}{R}$.

TABLE 1 Test Dataset based on random street topologies.

Topologies	n	m	Intersection density		Intersection popularity		Distance	
			min	max	min	max	min	max
Map1		50	342	1393	0	6	401	996
Map2	67	100	399	1388	0	9	433	939
Map3		250	360	1391	0	12	411	995
Map4		500	380	1301	0	18	422	944
Map5	72	350	560	1363	3	8	320	617
Map6		500	580	2388	3	11	346	577
Map7		600	610	2691	3	19	328	616
Map8		800	900	3600	3	35	337	580
Map9	224	600	1500	3393	5	18	360	697
Map10		750	1800	4000	5	24	389	657
Map11		900	1200	5000	5	30	369	696
Map12		1000	1500	7000	5	42	379	660

In order to show how the proposed algorithm works under different urban scenarios and to find the optimal number of RSUs in such areas, we compare the results obtained by our algorithm against three approaches. The first one is the greedy approach proposed by Chi et al. [10]. The second approach is a genetic intersection coverage algorithm (GICA) developed in our previous work [26]. The third one is a Heuristic Genetic Algorithm (HGA) proposed in section 3.

In each test, we have used an Ant colony consisting of 10 ants. The exploration rate was $q_0 = 0.1$, and the evaporation rates were $\varphi = 0.1$ and $\rho = 0.1$. For the influence factor of the heuristic, we used $\alpha = 5$. The initial value of the pheromone trail is$\tau_0 = 0.6$. Overall, 100 iterations were performed for each of the test sets associated to each road topology. For all topologies, we run the (GICA) and (HGA) algorithms with the following parameters:

$T = 100, p_{Cros} = 0.9, p_{Mut} = 0.01, \varrho = 80\%, \psi = 20\%, t = 100$ iterations.

We also analyzed the effect that different RSU transmission ranges and weights of the two traffic factors have on the network performance. In order to evaluate the effect of each traffic factor, we have distributed the weights of the two traffic factors contain vehicle density ($\boldsymbol{w1}$) and location popularity ($\boldsymbol{w2}$) in an interval [0, 1]. Table 2 illustrates the parameter settings of our experiments.

TABLE 2 Parameter settings and values.

Parameters	Values
RSU Transmission Range (R)	{250m, 350m,450m,550m}
Weights of factors (w_1, w_2)	(1,0) / (0.7, 0.3) / (0.5, 0.5) / (0.3, 0.7) / (0,1)

5.2. Experimental Results

Now, we present a set of experiments comparing the performance of the greedy algorithm proposed [10], GICA [26] and HGA algorithms against our proposed AC-RDV algorithm, considering the three classes of urban topologies defined earlier. Our goal is to quantify the impact of the transmission range, RSU deployment budget and traffic weight parameter through the coverage area ($\boldsymbol{Cov}$), average connectivity ($\boldsymbol{c_n}$), and overlapping area ($\boldsymbol{\delta}$) matrices. Therefore, we keep the total cost of RSU deployment under a predefined budget (number) and vary this budget in intervals $[10\%, , k]$.

5.2.1. Impact of the RSU transmission range

First, we have evaluated the total number of RSUs located, coverage area ($\boldsymbol{Cov}$), average connectivity ($\boldsymbol{c_n}$), and overlapping area ($\boldsymbol{\delta}$) for all topologies under test according to the RSU transmission range. The overlapping area is required to analyze the redundant duplicated traffic messages generated by the neighboring RSUs. For all instances, we have used the traffic parameters weight as $(\mathrm{w}_1, \mathrm{w}_2) = (0.7, 0.3)$. The numerical results are given in Tables 3, 4, 5 and 6.

In Table 3, for a transmission area range equals to 250 m, we observe that AC-RDV is still better than the other algorithms in both the RSUs number and solution quality ($\boldsymbol{Cov}, \boldsymbol{c_n}, \boldsymbol{\delta}$). We observe that the HGA gives better results compared to GICA, that means the heuristic initialization strategy (HGA) performs better than the random initialization strategy (GICA), this is because HGA starts with a population containing good solutions generated by a heuristic greedy approach.

TABLE 3 The numerical results for transmission range R=250 m

Map	**RSUs number**				**Average Coverage (%)**				**Average connectivity (%)**				**Overlap Rate (%)**			
	GA	***GICA***	***HGA***	***AC-RDV***	***GA***	***GICA***	***HGA***	***AC-RDV***	***GA***	***GICA***	***HGA***	***AC-RDV***	***GA***	***GICA***	***HGA***	***AC-RDV***
1	51	43	39	**39**	43,43	48,90	51,64	**52,55**	35,94	41,65	50,45	**51,65**	4,73	4,54	0,7	**0,4**
2	57	47	43	**41**	46,77	53,89	57,97	**58,99**	33,48	40,08	44,18	**47,33**	5,2	4,33	2,3	**1,2**
3	61	49	48	**45**	51,61	56,99	60,96	**62,03**	37,41	45,20	47,38	**50,71**	6,34	6,22	3,2	**2,62**
4	63	57	54	**49**	54,07	62,45	68,27	**68,47**	39,24	46,63	48,47	**53,20**	5,42	5,2	4,01	**3,66**
5	56	50	47	**42**	46,86	55,92	59,28	**62,81**	33,27	39,31	43,24	**48,00**	9,00	7,38	4,89	**2,05**
6	59	52	50	**46**	44,45	53,02	56,40	**59,68**	35,33	40,81	45,58	**49,92**	11,4	6,26	3,87	**1,96**
7	62	55	52	**47**	50,26	59,50	63,23	**66,47**	36,30	43,68	47,77	**51,34**	9,00	7,51	5,01	**3,52**
8	66	58	55	**50**	54,55	63,63	67,11	**72,04**	38,68	46,02	50,20	**55,76**	10,21	8,75	7,81	**3,11**
9	143	121	107	**98**	50,54	58,07	62,74	**65,16**	34,04	39,47	44,19	**49,34**	12,16	9,29	7,91	**4,07**
10	157	129	119	**104**	53,69	60,74	65,14	**67,28**	36,26	41,37	45,37	**50,49**	9,30	11,3	6,53	**4,48**
11	171	143	136	**111**	56,30	64,23	66,85	**69,77**	35,11	41,94	46,76	**51,24**	12,97	10,99	9,18	**6,03**
12	207	156	148	**131**	61,07	65,90	69,85	**72,73**	39,29	46,95	52,76	**57,20**	13,89	9,16	8,04	**7,31**

When the RSUs' transmission range is 350 m (see table 4), the difference between AC-RDV based coverage and HGA coverage becomes very small in first topology class. We can observe that HGA achieves a higher coverage than the GA, and it is slightly Upper that the coverage obtained by the GA algorithm. This can be explained by the characteristics of this traffic, which has less dense.

TABLE 4 The numerical results for transmission range R=350 m

Map	RSUs number				Average Coverage (%)				Average Connectivity (%)				Overlap Rate (%)			
	GA	*GICA*	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*
1	42	35	33	**27**	45,60	51,35	54,22	**55,18**	48,69	53,60	56,46	**62,11**	8,72	6,80	5,98	**3,59**
2	51	40	36	**31**	49,11	56,58	60,87	**61,94**	45,35	46,24	50,36	**55,40**	8,39	6,54	5,76	**3,46**
3	55	43	40	**35**	54,19	59,84	64,01	**65,13**	50,67	56,27	59,60	**65,56**	11,41	8,90	7,83	**4,70**
4	57	48	46	**38**	56,77	65,57	71,68	**71,89**	53,16	60,14	63,15	**69,47**	9,79	7,64	6,72	**4,03**
5	50	43	40	**32**	49,20	58,72	62,24	**65,95**	45,07	51,54	55,69	**61,26**	12,94	10,09	8,88	**4,95**
6	53	47	43	**35**	46,67	55,67	59,22	**62,66**	47,86	50,50	55,55	**61,11**	15,25	11,90	10,47	**5,83**
7	56	46	44	**36**	52,77	62,48	66,39	**69,79**	49,18	57,26	62,99	**69,29**	12,21	9,52	8,38	**4,67**
8	60	51	47	**39**	57,28	66,81	70,47	**75,64**	52,39	58,33	63,16	**69,48**	16,60	12,95	11,40	**6,35**
9	129	107	100	**77**	53,07	60,97	65,88	**68,42**	46,12	51,76	56,94	**62,63**	19,25	15,02	13,22	**6,93**
10	141	115	102	**80**	56,37	63,78	68,40	**70,64**	49,12	53,24	58,56	**64,42**	18,09	14,11	12,42	**7,46**
11	154	128	117	**87**	59,12	67,44	70,19	**73,26**	47,56	51,99	57,19	**62,91**	21,97	17,14	15,08	**9,05**
12	186	140	125	**102**	64,12	69,20	73,34	**76,37**	53,21	57,68	62,45	**68,70**	19,04	14,85	13,07	**7,85**

If we increase the extended range of RSU as R= 450 m (see table 5), our algorithm covers far more area and makes good network connectivity for a given number of RSUs compared to other approaches. It is obvious that increasing the wireless transmission range will have a significant impact on the average connectivity.

TABLE 5 The Numerical results for transmission range R=450 m

Map	RSUs number				Coverage (%)				Average connectivity (%)				Overlap Rate (%)			
	GA	*GICA*	*HGA*	*AC-RDV*	*GA*	GICA	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*
1	38	31	29	**24**	47,96	56,72	58,06	**69,68**	53,91	60,28	67,62	**73,34**	11,34	8,16	7,34	**5,51**
2	46	36	32	**28**	53,85	62,52	65,18	**65,09**	50,22	56,56	64,31	**67,19**	10,91	7,85	7,07	**5,30**
3	49	39	36	**31**	56,62	66,10	68,54	**68,45**	56,12	65,43	68,98	**72,00**	14,83	10,68	9,61	**7,21**
4	51	43	41	**34**	63,42	72,44	76,76	**75,55**	58,87	67,50	70,56	**75,53**	12,73	9,17	8,25	**6,19**
5	45	38	36	**29**	55,06	64,87	66,65	**69,31**	49,92	56,90	62,94	**68,15**	16,82	12,11	10,90	**8,18**
6	48	42	39	**31**	52,38	61,50	63,42	**65,86**	53,01	59,07	66,36	**70,87**	19,83	14,28	12,85	**9,64**
7	50	41	40	**32**	58,73	69,02	71,09	**73,34**	54,46	63,22	69,54	**72,89**	15,87	11,42	13,28	**9,96**
8	54	46	42	**35**	62,34	73,81	75,46	**79,49**	58,03	66,61	73,08	**79,17**	21,58	15,54	13,99	**10,49**
9	116	96	90	**69**	58,28	67,36	70,55	**71,90**	51,07	57,13	64,33	**70,05**	25,03	18,02	16,22	**12,17**
10	127	103	91	**72**	60,50	70,46	73,24	**74,24**	54,40	59,87	66,04	**71,68**	23,52	16,93	15,24	**11,43**
11	138	115	105	**78**	62,10	74,51	75,16	**76,99**	52,67	60,71	68,07	**72,75**	26,56	18,57	16,71	**12,53**
12	167	126	112	**92**	64,88	76,44	78,54	**80,26**	58,94	67,96	76,80	**81,21**	24,75	17,82	16,04	**12,03**

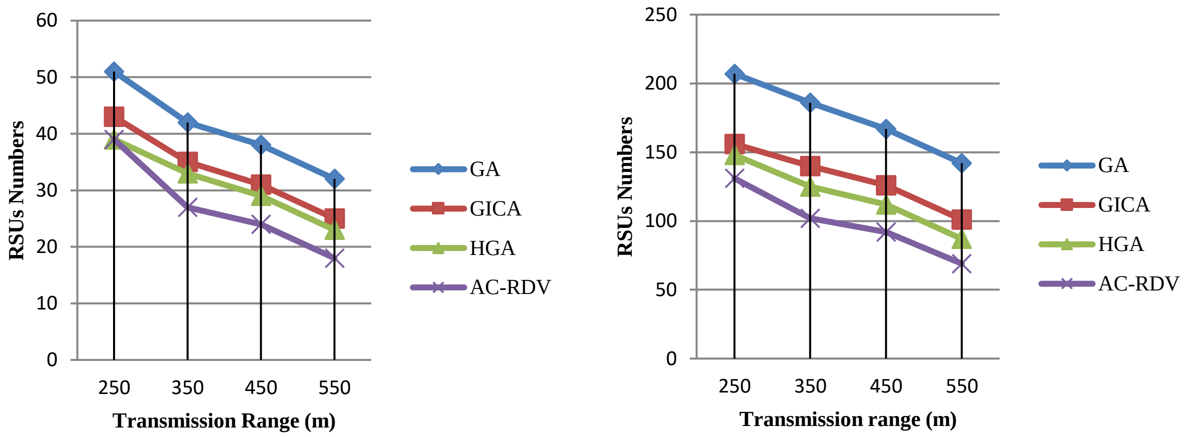
If we extend range the RSU as R= 550 m (see table 6), our algorithm covers much more area and makes good network connectivity for a given number of RSUs compared to other approaches. For example in the map 12 using the AC-RDV, the number of RSUs decreases to

27,92% with the average area coverage increasing to 19, 67 % and 38% as average connectivity, while the overlapping rate increases to reach 11,25% (see the last row of Table 6).

TABLE 6 The Numerical results for transmission range R=550 m

Map	RSUs number				Coverage (%)				Network connectivity (%)				Overlap Rate (δ) (%)			
	GA	*GICA*	*HGA*	*AC-RDV*	*GA*	GICA	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*	*GA*	*GICA*	*HGA*	*AC-RDV*
1	32	25	23	**18**	51,80	62,96	65,03	**80,13**	62,54	75,35	83,17	**88,74**	21,48	17,18	13,40	**11,39**
2	39	29	25	**21**	58,16	69,40	73,00	**74,85**	58,26	70,70	79,10	**81,30**	19,47	15,58	12,15	**10,33**
3	42	31	28	**23**	61,15	73,37	76,76	**78,72**	65,10	81,79	84,85	**87,12**	20,76	16,61	12,96	**11,02**
4	43	34	32	**26**	68,49	80,41	85,97	**86,88**	68,29	84,38	86,79	**91,39**	17,82	14,26	11,12	**9,45**
5	38	30	28	**22**	59,46	72,01	74,65	**79,71**	57,91	71,13	77,42	**82,46**	23,55	18,84	14,70	**12,50**
6	41	34	30	**23**	56,57	68,27	71,03	**75,74**	61,49	73,84	81,62	**85,75**	27,76	22,21	17,32	**14,72**
7	42	33	31	**24**	63,43	76,61	79,62	**84,34**	63,17	79,03	85,53	**88,20**	22,22	17,78	13,87	**11,79**
8	46	37	33	**26**	67,33	81,93	84,52	**91,41**	67,31	83,26	89,89	**93,80**	24,61	19,69	15,36	**13,06**
9	99	77	70	**52**	62,94	74,77	79,02	**82,69**	59,24	71,41	79,13	**84,76**	35,04	28,03	21,86	**18,58**
10	108	82	71	**54**	65,34	78,21	82,03	**85,38**	63,10	74,84	81,23	**86,73**	32,93	26,34	20,55	**17,47**
11	117	92	82	**59**	67,07	82,71	84,18	**88,54**	61,10	75,89	83,73	**88,03**	37,18	29,74	23,20	**19,72**
12	142	101	87	**69**	70,07	84,85	87,96	**92,30**	68,37	84,95	87,46	**95,26**	34,65	27,72	21,62	**18,38**

From the all results, we find that the average connectivity increases with the growth of vehicles density. Reducing the transmission range leads to keep only the vehicles behind the interaction connected in one big network partition that contains the majority of vehicles.



a) Map 1 b) Map 12

FIGURE 4 Number of RSUs required depending on the transmission range.

This clearly shows that our algorithm requires less number of RSUs for a given area, which makes the solution more economically reliable compared to the other approaches.

As it can be seen in Figure 4, increasing the RSU transmission range decreases the deployment cost. For the map 1 where the number of RSUs n= 67 and the number of road segments m= 50, for R=250 m to 550 m, the number of RSUs decreases into 31, 34% in AC-RDV. While, the map 12 contains 222 intersections and 1000 road segments, the number of RSUs decreases to 27, 93% in AC-RDV.

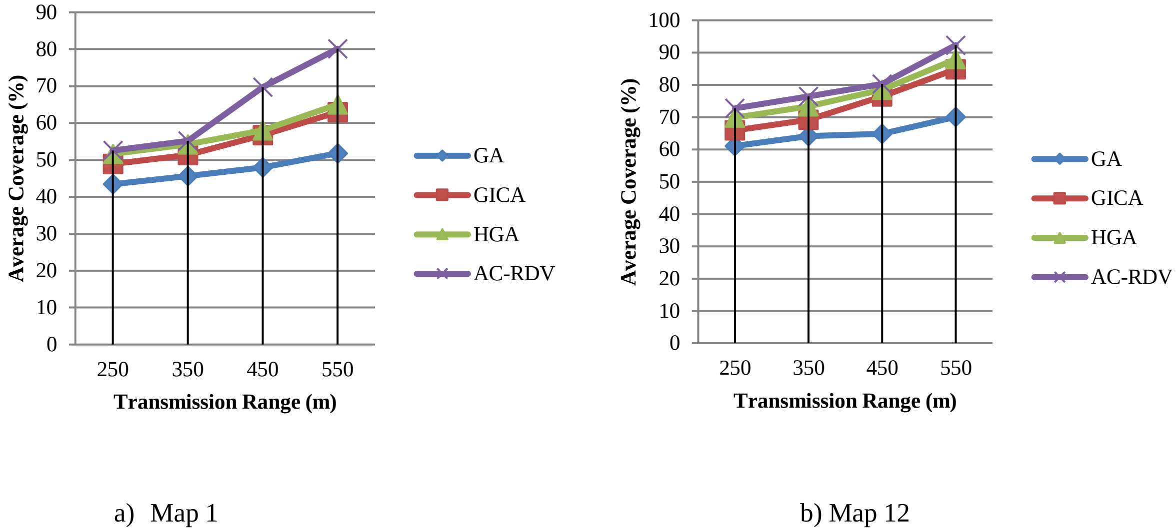
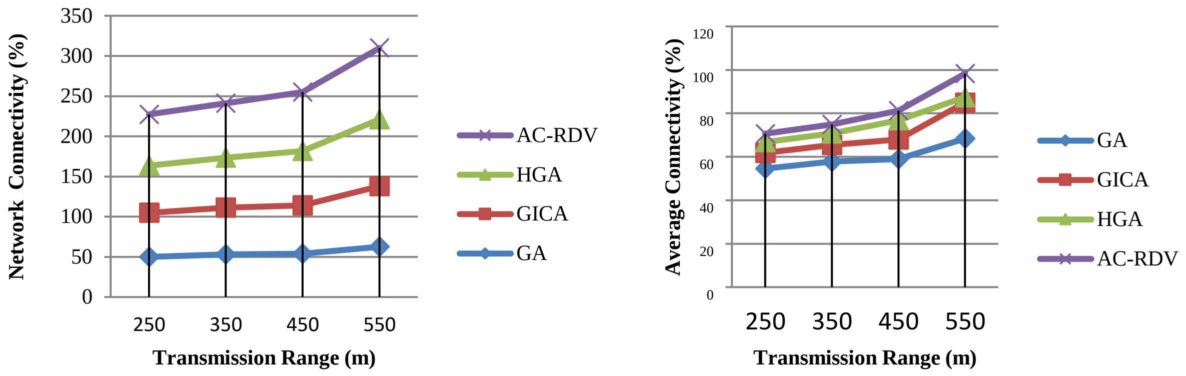


FIGURE 5 Average Coverage according to the RSU transmission range variation.

As shown in Figure 5, AC-RDV provides a good coverage average as the transmission range grows from 250m to 550m. In the map 1 the average area coverage increases to 27.58%, while the average area coverage in map 12 increases to 19.67 %. This is due to the distance between deployed RSUs, which is shorter than transmission area, which allows disseminating the message to RSUs. As for the 250m transmission range, AC-RDV based coverage also performs better than GA algorithm and GICA. Moreover, AC-RDV and HGA give the similar results for the map 1. As for the 250m transmission range, AC-RDV based coverage also performs better than GA algorithm and GICA. Moreover, AC-RDV and HGA give similar results as in map 1. This can suggest that, the effectiveness of our algorithm is appeared especially in the large-scale deployment. We have also investigated the impact of the transmission range on the connected intersections; we utilized the average connectivity as a metric.

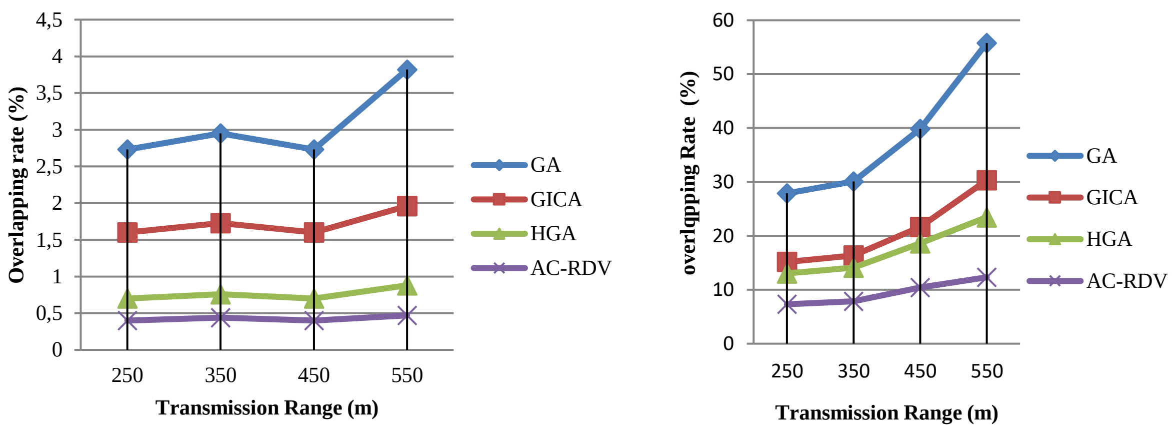
In Figure 6, AC-RDV remarkable the average connectivity is achieved though the transmission area is larger. Therefore, high transmission range is still needed to keep the network connected.



a)Map 1 b) Map

FIGURE 6 Average connectivity according to the RSU transmission range variation.

Similarly, Figure 7 shows that the overlapping ratio(δ) of each region, when using the AC-RDV algorithm, is quite lower than all other approaches. In this figures, we display the relationship between overlapping ratio (δ) and RSU transmission range of RSU(R). For $450 \leq R \leq 550$ m, we observe an increase of the overlapping ratio for the two neighboring intersections (see Figure 7), showing a proportionality relation between R and δ. Since the length of the road segments connecting two intersections I_i and I_j in all our topologies is in the range 401≤ L_{ij}≤ 996 m, the distance from the intersection i to intersection j is $d_{i,j} = 2 * R$, and 450≤ R ≤550 m, which explains the increase of the overlapping ratio for the two neighboring intersections (see Figure 5).



a) Map 1 b) Map

FIGURE 7 Overlapping rate when varying the RSU transmission range.

This situation explains that the transmission range of the RSU is proportional to overlapping ratio when $450 \leq R \leq 550$.

5.2.2. Impact of the RSUs Number

In order to know how well these RSUs are able to cover the network area, we fixed the deployment budget under to predefined number (K) of RSU. This k value can be measured as 30% of the number of intersections. In the considered topologies, thereafter, we test the variations in terms of coverage area, average connectivity, and the duplicate message transmission in each scenario. Indeed, we vary this number as the set {10%, 15%, 25%, 30%}. The Figure 8, 9, 10 summarizes the results for the map 10 using R=450m. Figure 8, as shwed, as the number of RSUs increases, so does the percentage of covered areas. Compared to the other approaches, AC-RDV improves the coverage area of RSUs under to less number of RSUs, which makes the solution more economically reliable. As for budget of deployment equals to 30 %, AC-RDV outperforms GA, GICA and HGA in terms of the average coverage by up to 34.9 %, 24.3%, and 15.7%, respectively. It is obvious that increasing the deployment budget will have a significant impact on the average connectivity.

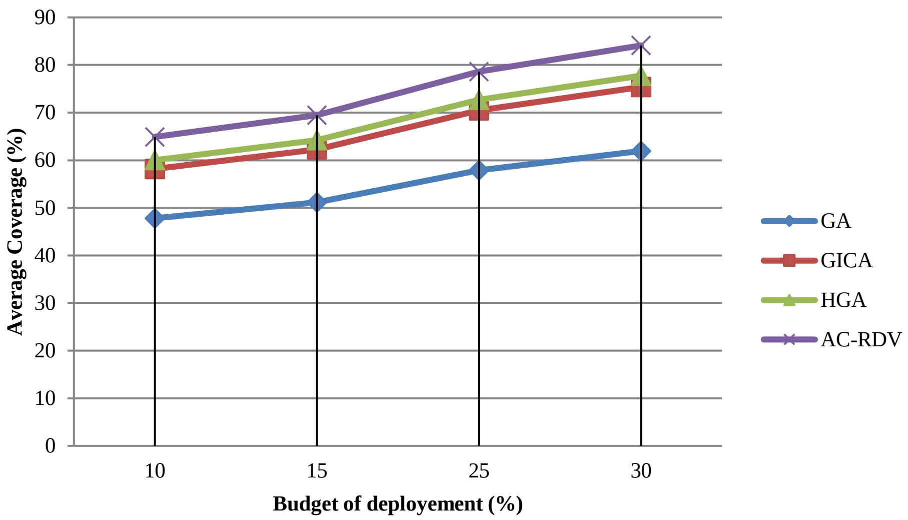


FIGURE 8 Average Coverage rate when varying the RSU number

From the figure 8, it can be seen that the more the number of RSUs increases, although coverage covers larger area of a road, which leads to large number of connected vehicles. We select a value k= 30% since the connectivity has been more affected by this number of RSUs. As can be seen in figure 8, the average connectivity provided by AC-RDV Algorithm for k=30% is more than a double of that insured by GA. Also, AC-RDV outperforms GICA, HGA by up to 22.9 % and 15.57%, respectively. We select a value k= 30% RSUs since the message coverage has been more affected by this number of RSUs.

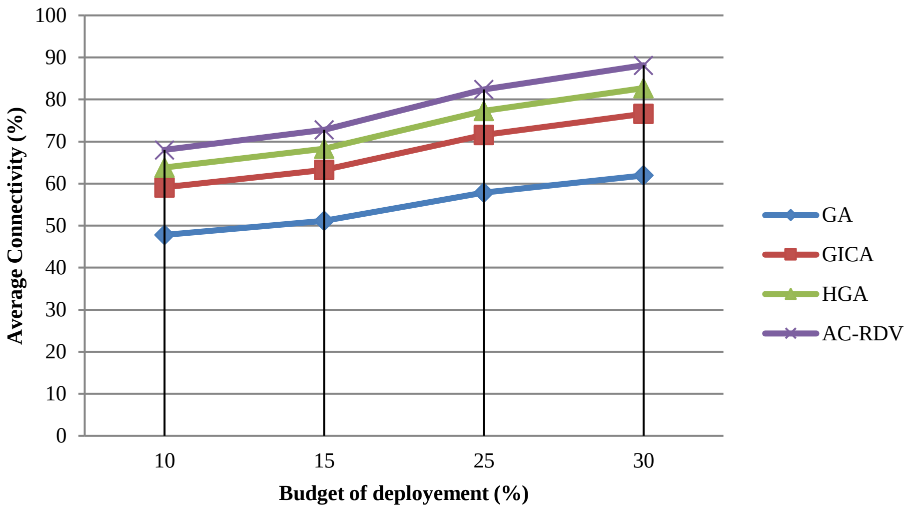


FIGURE 9 Average Connectivity according to the RSU number variation.

To decrease the redundant duplicated traffic messages generated by vehicles, it is required to analyze the overlapped area covered by two neighboring RSUs. However, aggressive retransmission may cause severe collisions. The results shown in figure 8, as the number of RSUs increases the overlapping rate increases. For a deployment cost from 10% to 30%, the overlapping rate increases to 3.82% (AC-RDV), 7.83% (HGA) , 8.60% (GICA), and 9.40% (GA).

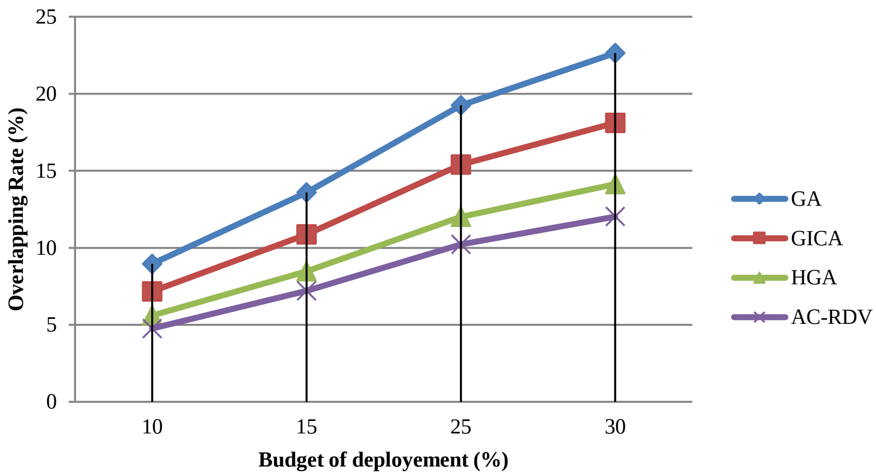


FIGURE 10 Overlapping rate when varying the RSU number .

5.2.3. *Impact of Weights on the traffic factors*

One says that an approach is stable if we can apply it using different criteria. To obtain the knowledge on how much the AC-RDV approach can be influenced by the weighs of the traffic factors, a set of tests were made where we changed the weights of the traffic factors (see Table 2). As shown in Figure10, the results of applying the four algorithms (GA, GICA, HGA and AC-RDV algorithm) on Map 4 prove that the greedy algorithm is more stable than the AC-RDV approach. This can be explained by the probabilistic aspect of our approach, since, in order to generate the solution, we use a stochastic transition rule. As a result, we can say that the change of weights does not influence our approach.

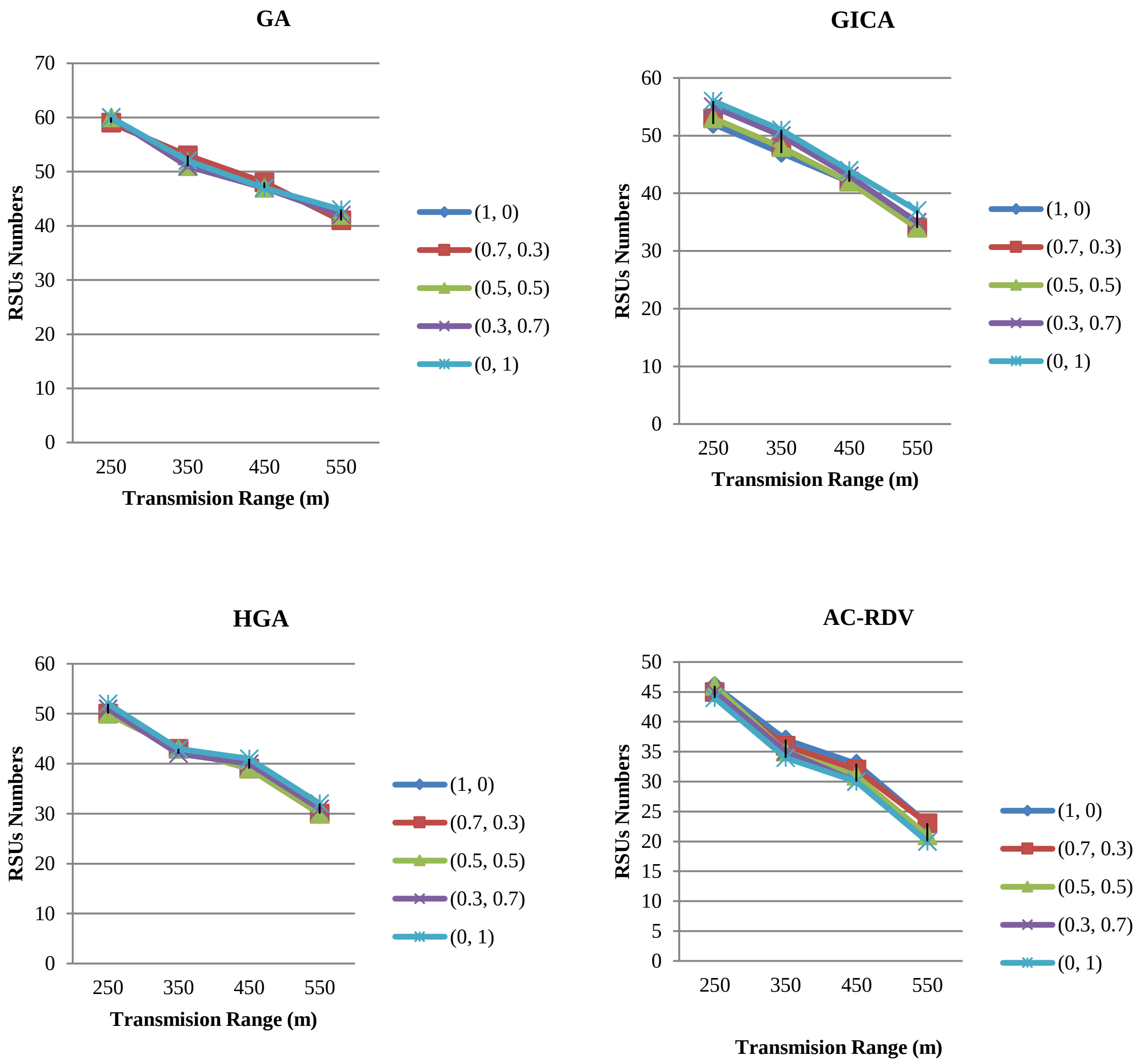


FIGURE 11 Impact of weights on the traffic factors on the RSU deployment

From this simulation study, we can conclude that our AC-RDV approach is a much better placement strategy than the greedy algorithm for urban vehicular networks in terms of the number of RSUs required and the area coverage achieved. To sum up, our approach is suitable for different traffic schemes for it significantly boosts the quality of communications in vehicular environments.

6. Conclusion

Dealing with the problem of RSU deployment in VANETs, we introduced in this study a new bio-inspired RSU placement system called "Ant colony optimization system for RSU deployment in VANET (AC-RDV)". AC-RDV is an intersection-coverage approach based on the intersection priority to deploy RSUs at the intersections having a higher impact on the efficiency of vehicular networks. Furthermore, AC-RDV provides a new dynamic heuristic function performed by considering the density of vehicles included in each time. For a more practical RSU deployment, based on graph model, we propose a vehicular network updating every time a new RSU is deployed. This could be achieved by removing the candidate intersections adjacent to the RSU when these intersections are located within its transmission range. We validated AC-RDC with extensive tests using different road topologies created randomly on various urban areas. Compared to the three approaches: GA, GICA, HGA, the reached results display that our scheme shows better performances in terms of reduced number of deployed RSUs and the overlapping ratio, while, to maximizing the coverage area and connectivity network.

As a further work, we intend to test our approach on large-scale urban environments based on realistic traffic traces. Moreover, a dynamic RSU deployment is hoped trying to put RSU in OFF mode for a certain period if there are vehicles that could replace the RSU to forward messages.

REFERENCES

[1] M.N. Mejri, J. Ben-Othman, M. Hamdi, Survey on security challenge and possible cryptographic solutions, Vehicular Communications, 1(2) (2018), pp. 53-66.

[2] S.M. Hanshi, T. Wan, M.M. Kadhum, A.A. Bin-Salem, Review of Geographic Forwarding Strategies for Inter-vehicular Communications from Mobility and Environment Perspectives, Vehicular Communications, 14 (2018), pp. 64-79.

[3] S. Bitam, A. Mellouk, Routing for vehicular Ad Hoc networks, Bio-Inspired Routing Protocols for Vehicular Ad Hoc Networks, Wiley Edition, 2014.

[4] M. Muhammad, G.A. Safdar,Survey on existing authentication issues for cellular-assisted V2X communication, Vehicular Communications, 12 (2018), pp. 50-65.

[5] Z. Wang, J. Zheng, Y. Wu, N. Mitton,A centrality-based RSU deployment approach for vehicular ad hoc networks. In : 2017 IEEE International Conference on Communications (ICC). IEEE, 2017. p. 1-5.

[6] H. Liu, S. Ding, L. Yang, T. Yang, A -based Strategy for Roadside Units Placement in Vehicular Ad Hoc Networks, International Journal of Hybrid Information Technology, 7(1) (2014), pp. 91-108.

[7] O. Trullols, M. Fiore, C. Casetti, C. Chiasserini, J. B. Ordinas, Planning roadside infrastructure for information dissemination in intelligent transportation systems, Computer Communications, 33(4) (2010), pp. 432 – 442.

[8] J. Hromkovič, Algorithmics for hard problems:introduction to combinatorial optimization, randomization, approximation, and heuristics, ed: Springer Science & Business Media, 2013.

[9] Y. Jo, J. Jeong, RPA: Road-Side Units Placement Algorithm for Multihop Data Delivery in Vehicular Networks, in: IEEE 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), (2016), pp. 262-266.

[10] J. Chi, Y. Jo, H. Park, S. Park, Intersection-priority based optimal RSU allocation for VANET, in: Fifth International Conference on Ubiquitous and Future Networks (ICUFN), (2013), pp.350–355.

[11] M. Dorigo, M. Birattari, C. Blum, L.M. Gambardella, F. Mondada, T. Stützle, Ant Colony Optimization and Swarm Intelligence, In: 6th International Conference (ANTS), Belgium, ed: Springer, 5217 (2004).

[12] Xu. Liya, H. Chuanhe, Li Peng, Z. junyu, A Randomized Algorithm for Roadside Units Placement in Vehicular Ad Hoc, in: IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks MSN, (2013), pp.193 – 197.

[13] C. Liu, H. Huang, H. Du, Optimal RSUs deployment with delay bound along highways in VANET, Journal of Combinatorial Optimization, 33(4) (2017), pp.1168-1182, 2017.

[14] B. Aslam, F. Amjad, C.C. Zou, Optimal roadside units placement in urban areas for vehicular networks, in: IEEE Symposium on Computers and Communications ISCC, (2012), pp.423–429.

[15] P. Patil, A. Gokhale, Voronoi-based placement of road-side units to improve dynamic resource management in Vehicular Ad Hoc Net-works, in: International Conference on Collaboration Technologies and Systems (CTS), (2013), pp.389–396.

[16] C. Ghorai, I. Banerjee, A constrained Delaunay Triangulation based RSUs deployment strategy to cover a convex region with obstacles for maximizing communications probability between V2I, Vehicular Communications, 13 (2018), pp. 89-103.

[17] E. S. Cavalcante, A.L. Aquino, G.L. Pappa, Roadside Unit Deployment for Information Dissemination in a VANET: An Evolutionary Approach, in: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation, ACM, New York, USA, (2012), pp. 27–34.

[18] J. F. M. Sarubbi, D. Vieira, E. Wanner, C.M. Silva, A GRASP-based heuristic for allocating the roadside infrastructure maximizing the number of distinct vehicles experiencing contact opportunities, in: IEEE Symposium on Network Operations and Management (NOMS), (2016), pp. 1187-1192.

[19] D. Kim, Y. Velasco, W. Wang , R. N. Uma, R. Hussain, S. Lee, A new comprehensive RSU installation strategy for cost-efficient VANET deployment, in: IEEE Transactions on Vehicular Technology, 66(5) (2017), pp.4200-4211,

[20] D. Irit, S. Safra, On the hardness of approximating minimum vertex cover, Annals of mathematics, (2005), p.p. 439-485.

[21] M. Dorigo, G.D. Caro, L.M. Gambardella, Ant Algorithms for Discrete Optimization, Artificial life, 5(2) (1999), pp. 137-172.

[22] M. Dorigo , L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, in: IEEE Transactions on Evolutionary Computation, 1 (1) (1997), pp. 53-66.

[23] A.B Reis, S. Sargento, F. Neves, Deploying roadside units in sparse vehicular networks: What really works and what does not, in: IEEE Transactions on Vehicular Technology, 63 (6) (2014), p.p. 2794-2806.

[24] R. Jovanovic, M. Tuba, An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem, Applied Soft Computing, 11(8) (2011), pp. 5360-5366.

[25] L. Lessing, I. Dumitrescu, T. Stützle, A comparison between ACO algorithms for the set covering problem, in: International Workshop on Ant Colony Optimization and Swarm, ed: Springer, Berlin, Heidelberg, (2004), pp. 1-12.

[26] A. Guerna, S. Bitam, Salim, GICA: An evolutionary strategy for roadside units deployment in vehicular networks,in: International Conference on Networking and Advanced Systems (ICNAS). IEEE, (2019). p. 1-6.

[27] PAPADIMITRIOU, Christos H. et STEIGLITZ, Kenneth. Combinatorial optimization: algorithms and complexity. Courier Corporation, 1998.

[28] S. Mandavilli et P. Lalit M. Genetic algorithms: A survey. computer, 1994, vol. 27, no 6, p. 17-26.