A GRASP Based Heuristic for Deployment Roadside Units in VANETs

João F. M. Sarubbi[*], Taís R. Silva [*], Flávio V. C. Martins[*], Elizabeth F. Wanner[*], Cristiano M. Silva[†]

[*]Departamento de Computação, Centro Federal de Educação Tecnológica-MG, Brazil
[†]Departamento de Computação, Universidade Federal de São João Del-Rei, Brazil
Emails: joao@decom.cefetmg.br, silva.taisrocha@gmail.com, flaviocruzeiro@decom.cefetmg.br
efwanner@decom.cefetmg.br, cristiano@ufsj.edu.br

***Abstract*—In this work we propose a new algorithm, Delta-r-GRASP, for solving the allocation of Roadside Units (RSUs) in a Vehicular Network. Our goal is to find the minimum set of RSUs to meet a Deployment $\Delta_{\rho_2}^{\rho_1}$. The Deployment $\Delta_{\rho_2}^{\rho_1}$ is a metric for specifying minimum communication guarantees from the infrastructure supporting the Vehicular Network. We compare our algorithm with a baseline algorithm, Delta-r. Moreover, we compare our results with the optimal value achieved by solver CPLEX. Our results demonstrate that our approach requires up to 85% fewer RSUs to achieve the same deployment efficiency, and our results differ no more than 15% from the optimal values.**

I. Introduction

The study on Vehicular Networks [1] (VANETs) constitutes a significant research segment and has been received a remarkable attention in the last years. We attributed such importance to the fact that VANETs - a particular type of mobile network, designed to the domain of vehicles and pedestrians - play a central role in Intelligent Transportation Systems (ITS). The first reason for the development of these networks was the traffic safety [2]. However, there are other VANETs applications such as: (i) video delivery [3]; (ii) monitoring of vehicles [4]; (iii) monitoring road conditions [5]; (iv) mobile infotainment [6]; (v) collaborative driving [7]; and so forth.

In a Vehicular Network, the communication may happen in two major ways: (i) Vehicle-to-Vehicle (V2V) [8] and (ii) Vehicle-to-Infrastructure (V2I) [3]. Figure 1 represents a Vehicular Network.

In V2V network, the communication occurs without any support infrastructure (pure ad hoc) and the communication is performed from vehicle to vehicle. In V2I network, the communication occurs through connections between vehicles and communication units, called Roadside Units (RSUs). The RSUs are fixed infrastructure positioned along pathways. Figure 2 represents a general V2I communication. Although V2V network does not need a support infrastructure, the communication can become inefficient in sparse areas, rural zones and low peak hours due to the lack of vehicles [8]. Although a V2I network can improve the general efficiency of a Vehicular Network, the main drawback of this network is the cost to install RSUs, turning the decision of RSUs number and location a challenge to network providers [9]–[13]. In this work, we try to provide an efficient strategy to one of the recurring problems in Vehicles-to-Infrastructure (V2I) networks: *Where to install the RSUs to minimize the number of RSUs and also ensure a minimum Quality of Service (QoS) to the general population?*

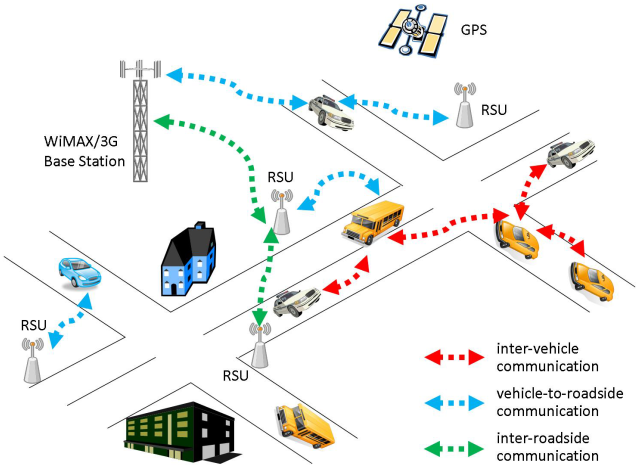


Figure 1: Vehicular Network

To answer this question, we must choose a metric to measure the QoS in Vehicular Networks. In 2015, Silva and Meira [3] proposed a new metric to measure the QoS in Vehicular Networks called Delta Network. In this metric, the QoS is measured using two different perspectives:

1) the individual user (ρ_1): wants to stay more time connected as possible or, at least, staying connected for a sufficient time to receive the desired information;
2) the traffic authorities (ρ_2): want that, at least, a fraction of the vehicles receives the desired information.

The individual user perception is entirely dependent upon the application. For traffic and time monitoring, the vehicle can receive information from time to time. On the other hand, for music and video streaming, the vehicle must receive a more "continuous" communication. In other words, Delta Network is based on two measurements:(i) connectivity duration; and, (ii) percentage of vehicles presenting such connectivity duration.

In this work, we propose a new algorithm for solving

978-3-901882-89-0 @2017 IFIP

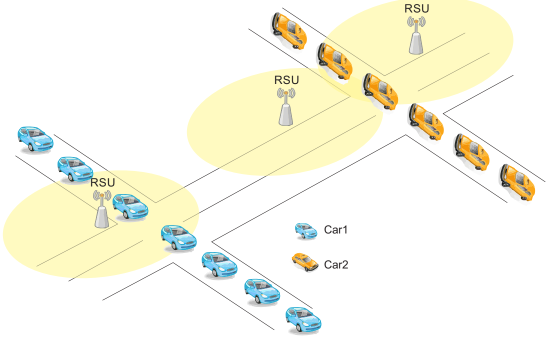


Figure 2: V2I Network

the allocation of RSUs to guarantee a Delta Network. Our algorithm, called Delta-r-GRASP, is based on the GRASP metaheuristic [14]. We choose this metaheuristic because it has been used successfully with other combinatorial optimization problems. Our goal is to find the minimum set of urban cells U where ρ_2 percent of the vehicles are ρ_1 percent of its travel time connected. The algorithm uses, as the procedure to build the Restricted Candidate List (RCL), a variation of Delta-r algorithm proposed by Sarubbi and Silva [15]. Besides, we compare our solution to the optimal value achieved by solver CPLEX.

Our results demonstrate that:

- Our algorithm can reduce the number of RSUs by more than 80% when compared to Delta-r algorithm;
- In all tested instances that we know the optimal value, our algorithm solution is no more than 15% from the optimal value;
- The use of a local search procedure in a greedy algorithm as Delta-r can reduce the number of RSUs up to 72%.

This work is organized as follows: Section II explains the used metric. Section III presents a selection of related work. Section IV formalizes the Deployment $\Delta_{\rho_2}^{\rho_1}$. Section V presents our proposal to represent complex road network. Section VI presents our baseline algorithm. Section VII presents our proposed solution. Section VIII presents our experiments. Section IX concludes our work.

II. The Deployment $\Delta_{\rho_2}^{\rho_1}$

The concept of Deployment $\Delta_{\rho_2}^{\rho_1}$ was initially proposed in Silva and Meira [3]. In this metric, the QoS is measured by two parameters (ρ_1, ρ_2). The first one, ρ_1, is a connection duration factor denoting how long each vehicle must stay connected to belong to the solution. The parameter ρ_1 is relative to the total travel time of each vehicle. For instance, if the network provider wants each vehicle to remain connected to the RSUs during 20% of its trip, ρ_1 must be set to 0.2. The second one, ρ_2, denotes the percentage of vehicles (from the total number of vehicles) must experience the connectivity defined by ρ_1. Thus, a $\Delta_{\rho_2}^{\rho_1}$-Deployment must guarantee that ρ_2 percent of the number of vehicles must be connected during (at least) ρ_1 percent of its trip. For instance, a deployment is $\Delta_{0.3}^{0.2}$ if 30% of the vehicles are connected to 20% of its trip duration.

In such manner, depending upon the required application, the network provider can choose different values for ρ_1 and ρ_2 parameters. Thus, the $\Delta_{\rho_2}^{\rho_1}$-Deployment can find the best locations to install the RSUs to achieve the expected QoS minimizing the number of RSUs. Besides, we must remember that the $\Delta_{\rho_2}^{\rho_1}$-Deployment is only a metric. It does not specify how the QoS is achieved because it is technology-independent. According to Silva and Meira [3], ”the metric does not care for what kind of access technology (Wi-Fi, 4G, Bluetooth, etc.) is used to perform the communication”.

Figure 3 illustrates the $\Delta_{\rho_2}^{\rho_1}$ metric. Differently from classical approaches, the metric is not represented by a single value. Instead, Delta is represented as a curve in a 2D plan. The x-axis indicates ρ_1, while the y-axis indicates ρ_2. In fact, Delta is the relation between ρ_1 and ρ_2.

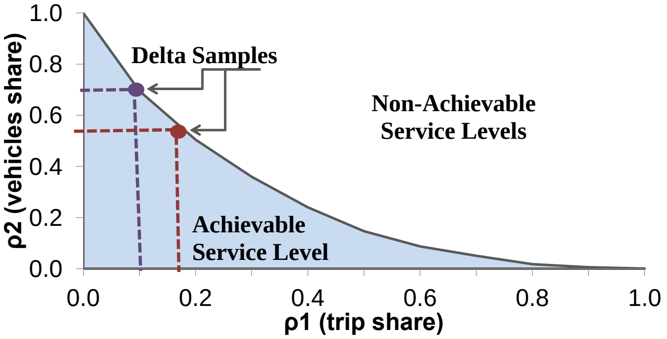


Figure 3: Delta is a curve that rules the combinations of ρ_1 and ρ_2.

It is important to note the Deployment $\Delta_{\rho_2}^{\rho_1}$ can be used with V2I and also V2V communication. The seminal work of Silva and Meira [3] also shows the importance to use a hybrid network with V2I and V2V. The hybrid communication can reduce the number of RSUs to achieved the same quality of service because the communication between vehicles helps in the overall solution. However, in this work, we will deal just with the V2I communication. Our objective is to measure the efficiency of the proposed algorithm and compare with other articles that use just the V2I network.

III. Related Work

For infrastructure deployment, Alpha Coverage [16] minimizes the number of Roadside Units ensuring that each path of length α from a road network must have at least one RSU. The contact probability is also considered: Zheng et al. [17] present the evaluation of a deployment strategy through the contact opportunity measuring the fraction of distance (or time) that a vehicle is in contact with the infrastructure. Lee and Kim [18] propose a greedy heuristic to place the roadside units aiming to improve vehicles connectivity while reducing disconnections. The heuristic counts the number of reached vehicles at each intersection considering the transmission range of the roadside units. Trullols et al. [19] formulates the allocation of roadside units as a Maximum Coverage

Problem [20]. Nekoui et al. [21] propose the definition of an infrastructure for Vehicular Networks based on the conventional definition of the transport capacity.

Barrachina et al. [12] present three RSUs deployment policies: (i) the Minimum Cost, that considers just the cost to install the RSUs. This strategy prioritizes locations that already have Internet access leaving that some areas remain isolated. (ii) the Uniform Mesh, that consist on distributing RSUs uniformly on the map. This strategy reduces the probability of having shadow areas in the map but not taking into account the real flow of vehicles traveling around the city. (iii) the D-RSU deployment, where "RSUs are placed using an inverse proportion to the expected density". The authors consider that vehicles can use V2V communication, and the RSUs are more important in low-density areas.

Barrachina et al. [22] present an architecture to estimate traffic density that combines V2V and V2I communication. They use a roadmap topology features from real cities and uses a ns-simulator to estimate the traffic. Kchiche and Kamoun [13] use a centrality and equidistant-based (uniform) deployment to optimize the delay and ensure a regular and stable service in a V2I and V2V network. The use of existing network infrastructures is also investigated: Marfia et al. [23] propose the use of open Access Points. Tonguz and Viriyasitavat [24] propose the utilization of vehicles as roadside units by using a biologically inspired network.

Optimization models for the deployment are also presented. Cruces et al. [25] introduce a mixed-integer quadratic programming based on optimum roadside units deployment scheme to provide Internet access services for the maximum road traffic volumes with a limited number of roadside units. Aslam et al. [26] present an Integer Programing Formulation (IPF) for choosing the best place to allocate RSUs. Sarubbi and Silva [15] present an Integer Programing Formulation for the Deployment $\Delta^{\rho_1}_{\rho_2}$ but the authors do not present any result using the proposed model.

For the specific Deployment $\Delta^{\rho_1}_{\rho_2}$ only three works present algorithms for this problem. The first one was Silva and Meira [3] that present the Delta-g algorithm. The Delta-g algorithm uses the following strategy: it selects urban cells using a greedy choice based on the **absolute** contact time provided by each urban cell when covered by a roadside unit. For each urban cell, Delta-g computes the sum of times of all uncovered vehicles that cross each urban cell and iteratively selects the urban cell that presents the highest sum. Silva and Meira [3] compare the Delta-g algorithm with DL algorithm proposed in Trullols et al. [19]. DL algorithm works as follows: while the share of ρ_2 covered vehicles is not achieved, the heuristic iteratively selects the densest urban cell still not having a roadside unit. The selected urban cell is added to the solution set, and DL recomputes the number of ρ_1-covered vehicles. After, Sarubbi and Silva [15] present a greedy relative contact time approach called Delta-r and compare with DL algorithm [19] and Delta-g algorithm [3]. As showed in Sarubbi and Silva [15] work, the Delta-r algorithm seems to be a better option when compared with Delta-g and DL algorithms. More recently, Sarubbi et al. [27] present a genetic algorithm for this problem with good results. The authors use a variation of Delta-r algorithm to create the initial population. However, they present solutions only for three pairs of ρ_1 and ρ_2.

IV. Problem Definition - Deployment $\Delta^{\rho_1}_{\rho_2}$

A Deployment is $\Delta^{\rho_1}_{\rho_2}$ whenever ρ_2 percent of all vehicles must be connected to roadside units during ρ_1 percent of the trip. Formally:

[Deployment $\Delta^{\rho_1}_{\rho_2}$] Let R represent a road network, and $V = \{v_1, v_2, \ldots, v_n\}$ represent the set of vehicles traveling R. Let the collection $T = \{U_1, U_2, \ldots, U_n\}$ represent the trajectory for each vehicle $v \in V$. Thus, each $v_k \in V$ is assigned a trajectory $U_k \in T$. Each $U \in T$ represent a set of urban cells $U_k = \{u^k_1, u^k_2, \ldots, u^k_m\}$ crossed by vehicle v_k during the trip. Let $C \subset V$ hold vehicles v_k experiencing percentage of connection $\geq \rho_1 \; \forall \; u \in U_k$. A deployment is considered $\Delta^{\rho_1}_{\rho_2}$ whenever $\frac{|C|}{|V|} \geq \rho_2$.

This problem can also be defined by a linear integer formulation. Suppose the set of vehicles V where $V = \{1, 2, ..., k\}$ and the set of urban cells U where is possible to put a roadside unit where $U = \{0, 1, ..., u\}$.

We have the following set of binary variables: a_u is equal to one if the urban cell u is chosen to belong to the solution and is zero otherwise; v_k is equal to one if the vehicle k belongs to the solution and zero otherwise. We also have the parameter t_{uk} that represents the time the vehicle k remained in urban cell u and tv_k that represents the total travel time of vehicle k.

The mathematical model is given by:

$$min \sum_{u \in U} a_u \qquad (1)$$

$$\sum_{u \in U} (t_{uk}/tv_k) a_u \geq \rho_1 v_k \quad \forall k \in V \qquad (2)$$

$$\sum_{k \in V} v_k \geq \rho_2 \, |V| \qquad (3)$$

$$a_u \in \{0, 1\} \quad \forall u \in U \qquad (4)$$

$$v_k \in \{0, 1\} \quad \forall k \in V \qquad (5)$$

Objective function (1) consists of minimizing the number of roadside units. Constraints (2) guarantee that a vehicle is chosen to belong to the solution only if it is connected $\rho_1\%$ of its travel time. Constraint (3) ensure that a minimum number of the vehicles is chosen to belong to the solution. Constraints (4) and (5) are the integrality constraints.

Delta Deployment can be considered NP-hard because it can be reduced to the Set Cover Problem (SCP) [28]. In the SCP we have different sets, (e.g. A, B, C) and each set has several elements. The principal goal is to find the minimum number of sets to cover all elements contained in the sets. The Delta Deployment can seem like a generalization of the SCP which the sets represent the urban cells and the elements of each set are the vehicles that cross the particular urban cell. Then, if ρ_1 is greater than zero and rho_2 is equal to 100%, then Delta Deployment becomes the Set Cover Problem [28].

V. Representing Road Networks

In this work, instead of using the original road network, we partitioned the urban area into a set of adjacent same size cells (i.e., grid model) and, once the city or region is partitioned, we abandon the original road network. This strategy was also used in Sarubbi and Silva [15] and Sarubbi et al. [27] The major advantages of this approach are: (i) the possibility to be more/less accuracy just increasing/decreasing the number of grid cells inside the region; (ii) the opportunity to reduce the computational efforts reducing the number of possible locations to install RSUs; and, (iii) the complexity of the solution that is not depending upon the flow and works in the same manner for big and small regions.

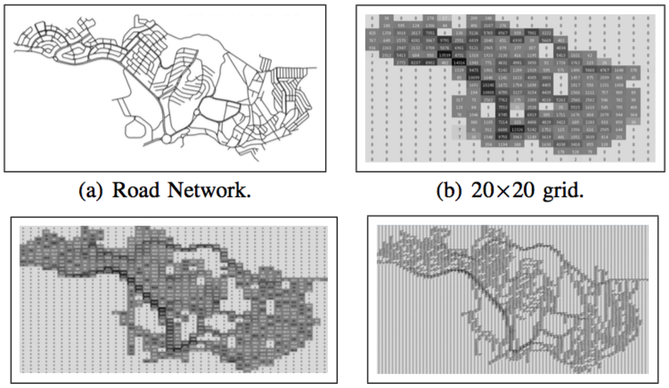

(a) Road Network.
(b) 20×20 grid.
(c) 40×40 grid.
(d) 80×80 grid.

Figure 4: Distinct Grid Setups.

Figure 4(a) shows a real road network (Ouro Branco city, Brazil). Figures 4(b) to 4(d) show how such road network may be modeled by grid setups from 20×20 up to 80×80.

VI. Baseline Algorithm

In this section, we present our baseline algorithm, called Delta-r, to solve the Deployment $\Delta^{\rho_1}_{\rho_2}$. Delta-r was proposed by Sarubbi and Silva [15] and was compared with DL algorithm [19] and Delta-g algorithm [3], presenting better results when compared with both previous algorithms.

Delta-r receives as input the matrix M describing the density of vehicles along the entire partitioned road network, the set V of vehicles, the collection T of trajectories, and the QoS parameters (ρ_1, ρ_2). The algorithm uses the following strategy: it selects urban cells using a greedy choice based on the **relative** contact time provided by each urban cell when covered by a roadside unit. For each urban cell, Delta-r calculates the percentage contribution obtained by deploying a roadside unit covering the given urban cell. The percentage contribution is based on the total trip time of each vehicle. For instance, if a vehicle k travels during 60 seconds and crosses the urban cell u during 30 seconds the score of the given urban cell associated with this particular vehicle is 0.5 representing the value 50%. In order to obtain the score of each urban cell, the algorithm simply counts the percentage contribution of all vehicles that not reached the ρ_1 criterion and crossing that urban cell. The algorithm iteratively selects the urban cell presenting the highest sum until $\rho_2\%$ of all vehicles are covered. The algorithm 1 presents a pseudo-code of the Delta-r algorithm proposed by Sarubbi and Silva [15].

Algorithm 1: Delta-r

```
Data: $M, V, T, \rho_1, \rho_2$
$Solution \leftarrow \emptyset$;
while $\frac{|C|}{|V|} < \rho_2$ do
    $\varphi \leftarrow$ Cell_Max_Relative_Time($M, V - C$);
    $Solution \leftarrow \varphi$;
    $M \leftarrow M - \varphi$;
    $C \leftarrow$ Connect($M, V, T, Solution, \rho_1$);
end
return $Solution$;
```

VII. Proposed Algorithm

In this section, the new proposed strategy to solve the Deployment $\Delta^{\rho_1}_{\rho_2}$, Delta-r-GRASP, is presented. The Delta-r-GRASP algorithm is based on the well-known Greedy Randomized Adaptive Search Procedure (GRASP) [14]. This algorithm is a multi-start metaheuristic that consists of two phases: (i) Construction Phase; (ii) Local Search Phase. Both phases are repeated for each iteration.

The Construction Phase consists of a randomized greedy function building up an initial solution. The solution is then used in the Local Search. The final result is simply the best solution found over all iterations. Algorithm 2 present the general Delta-r-GRASP algorithm.

Algorithm 2: Delta-r-GRASP

```
Data: $M, V, T, \rho_1, \rho_2, \alpha, Max\_Iterations$
for $it \leftarrow 1$ to $Max\_Iterations$ do
    Construction_Phase($M, T, V, \alpha, \rho_1, \rho_2$);
    Local_Search($M, T, V, \rho_1, \rho_2$);
end
```

A. Construction Phase

At the Construction Phase, a randomized greedy technique provides feasible solutions. Each feasible solution is iteratively constructed, one element at a time. However, instead of always selecting the best solution, a Restricted Candidate List (RCL) of good elements is built, and one element (not necessarily the top candidate) is randomly selected. Algorithm 3 presents the proposed Construction Phase algorithm.

We use, as a Build_RCL Procedure, a variation of the Delta-r algorithm [15]. However, instead of choosing the urban cell with maximum **relative** contact time, as the original Delta-r algorithm, we create a list of urban cells with the

```
Algorithm 3: Construction Phase
  Data: $M, T, V, \alpha, \rho_1, \rho_2$
  $Solution \leftarrow \emptyset$;
  while $\frac{|C|}{|V|} < \rho_2$ do
      Build_RCL($M, T, V, \alpha, \rho_1$);
      Selected_Cell $\leftarrow$ Random_Element($RCL$);
      Solution $\leftarrow$ Solution $\cup$ Selected_Cell;
  end
  return $Solution$;
```

best **relative** contact time and randomly select one element of the list. An RCL parameter α that can vary from 0.0 to 1.0, determines the level of greediness or randomness at the Construction Phase. When $\alpha = 0.0$, the Construction Phase becomes the original Delta-r algorithm, a simple greedy algorithm. Otherwise, when $\alpha = 1.0$, the Construction Phase becomes totally random. If the RCL is built with many elements, then many different solutions will be produced, according to chosen α value. The Algorithm(4) presents a pseudo-code from the Build_RCL Procedure.

```
Algorithm 4: Build_RCL
  Data: $M, T, V, \alpha$
  $RCL \leftarrow \emptyset$;
  CellMin $\leftarrow$ Select_Cell_Min_Rel_Contact_Time();
  CellMax $\leftarrow$ Select_Cell_Max_Rel_Contact_Time();
  Min $\leftarrow$ Select_Relative_Contact_Time(CellMin);
  Max $\leftarrow$ Select_Relative_Contact_Time(CellMax);
  foreach Remaning UrbanCell u do
      if Relative_Contact_Time(u)
        $\leq [Max - \alpha(Max - Min)]$ then
          RCL $\leftarrow$ RCL $\cup$ u;
      end

  end
  return $RCL$;
```

The selection process of candidate elements is determined by the rank of all items, according to their greedy function values. A good element is so designed because it belongs to the set of well-ranked elements.

The Build_RCL Procedure works as follows: first, the RCL list is set to empty. Then we compute the Maximal and Minimal relative contact time. For each remaining urban cell, we verify if the relative contact time of this cell is less than $Max - \alpha(Max - Min)$. If the condition is true, we add this urban cell to the RCL list.

B. Local Search Phase

After receiving the RSUs locations chosen by the Construction Phase, the Local Search algorithm finds a local optimum according to a chosen neighborhood. The Local Search algorithm works iteratively, replacing the current solution for a better one belonging to the neighborhood of the current solution. The Local Search Phase ends when it is not possible to find a better solution.

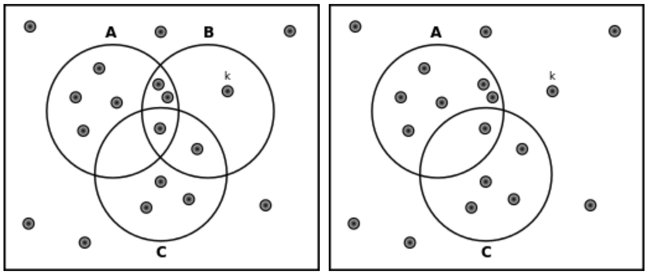
The local search procedure involves fundamental questions of the project, among which stands out the definition of the neighborhood and the search strategy in the vicinity. Regarding Deployment $\Delta_{\rho_2}^{\rho_1}$, the solution set is determined as a set of cells in which communication units will be positioned in agreement with the parameters ρ_1 and ρ_2 established. The neighborhood of that solution, in turn, consists of sets of cells which differ in a communication unit to meet the same parameters determined by $\Delta_{\rho_2}^{\rho_1}$.

The strategy defined for the local search originated from the perception that it is possible, from the solution set generated by the construction phase, to remove one or more RSUs, so that the criteria defined by Deployment $\Delta_{\rho_2}^{\rho_1}$ are still guaranteed. Two situations allow removing an RSU:

1) All vehicles covered by the current RSU are sufficiently served by other RSUs, keeping the condition set by the parameter ρ_1.
2) The current RSU does not cover some vehicles. However, it is possible that even with less covered vehicles the condition set by parameter ρ_2 is not violated.

The Figure 5 present an example of three RSUs (A, B, and C) that are communicating with twelve vehicles. After removing the RSU B that is communicating with five vehicles, the remaining RSUs (A and C) maintains communication with eleven vehicles because four of the five vehicles are communicating with others RSUs. Formally, we can remove the RSU B if one of the following conditions is met:

1) $(A \cup B \cup C) - (A \cup C) = \emptyset$;
2) $|(A \cup B \cup C) - (A \cup C)| \leq |(A \cup B \cup C)| - \rho_2 |V|$;



(a) Before Local Search **(b)** After Local Search

Figure 5: This Figure shows 3 RSUs (A, B and C) represented by the big circles and some vehicles representing by the small circles. If, for instance, a small circle k is within the big circle B means that the vehicle k is communicating with RSU B.

VIII. Experiments

In this section, we present experiments comparing Delta-r, Delta-r-GRASP and the ILP presented in section IV using solver CPLEX. Experiments are based on the realistic mobility trace (http://kolntrace.project.citi-lab.fr/)of Cologne, Germany. The trace is composed of 7,200s of traffic from 75,515 vehicles. All experiments are performed using the SUMO

(Sumo Simulator: http://sumo-sim.org) simulator and a set of tools designed by our team. SUMO runs the Cologne scenario and outputs the location of each vehicle (our mobility trace T) over time. The Partition Program reads the mobility trace, computes the bounding box of the mobility trace, partitions the Cologne into a grid of $\psi \times \psi$ urban cells, and then translates the mobility trace from Cartesian coordinates to Grid coordinates. For all experiments we use $\psi = 100$, resulting in about covered area of $260m \times 260m$ for each urban cell.

We present two sets of experiments. In the first one, we compare Delta-r algorithm proposed by Sarubbi and Silva [15] and Delta-r-GRASP algorithm with solver CPLEX. In the first set, we use an instance with 100 vehicles. Our goal is to measure how far is the solution achieved by Delta-r-GRASP algorithm from the optimal value. In the second one, we compare Delta-r-GRASP with the baseline Delta-r algorithm using an instance with 75,515 vehicles. Our objective is to measure the gain of Delta-r algorithm for an instance that is not possible to find the optimal value using solver CPLEX.

A. Solving the 100 vehicles instance

In this section, we compare the number of RSUs found by Delta-r algorithm with the optimal value achieved by solver CPLEX using the realistic mobility trace of Cologne, Germany. Since finding the solution using the mathematical model presented in Section IV has exponential complexity, we are not able to find the optimal value when we use the entire Colognes mobility trace (composed of 75,515 vehicles). Then, in this section, we present a study considering just the 100 first vehicles of the mobility trace.

For all instances, we run Delta-r-GRASP algorithm 11 times for seven different α values. We use $\alpha = \{0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and present the results for the set with 11 solutions with the minimal number of RSUs. For all experiment the $Max\_Iterations$ parameter was set to 5000.

Table I presents some results for 25 pairs of ρ_1 and ρ_2. The field *DR* presents the number of RSUs found by our baseline algorithm - Delta-r. The field *DR-LS* presents the number of RSUs found when we run the proposed local search after simple Delta-r algorithm. In other words, DR-LS algorithm is the Delta-r-GRASP algorithm with $\alpha = 0$ and $Max\_Iterations = 1$. Our objective is to measure the influence of the proposed Local Search at the final solution. To better represent the performance and the robustness of our algorithm Delta-r-GRASP, we present the best solution ($q_{0.00}$), the worst solution ($q_{1.00}$), and the median solution ($q_{0.50}$) amongst the 11 solutions used for each instance.

The Figure 6 presents the gain, in percentage, between CPLEX, DR-LS, and Delta-r-GRASP algorithms compared with Delta-r algorithm for 3 different ρ_1: (a) $\rho_1 = 0.1$;(b) $\rho_1 = 0.5$; (c) $\rho_1 = 0.9$. As we can note in Figure 6, we have gains up to 81% and, the use of a simple local search procedure can represent gains up to 72%. Besides, our proposed Delta-r-GRASP algorithm is, at some times, very close to the optimal solution.

Table I: Delta-r-GRASP Results - 100 Vehicles Instance

ρ_1	ρ_2	DR	DR-LS	Opt	DR-GRASP		
					$q_{0.00}$	$q_{0.50}$	$q_{1.00}$
0.1	0.1	5	4	**3**	**3**	3	3
0.1	0.2	9	8	**7**	**7**	8	8
0.1	0.3	14	13	**12**	**12**	12	13
0.1	0.4	20	20	**17**	**17**	18	19
0.1	0.5	28	26	21	23	25	26
0.3	0.1	18	12	**10**	**10**	11	12
0.3	0.2	33	26	20	22	23	23
0.3	0.3	56	36	32	33	36	39
0.3	0.4	71	53	45	49	50	51
0.3	0.5	88	70	58	64	67	68
0.5	0.1	44	20	**16**	**16**	18	19
0.5	0.2	92	49	38	41	43	44
0.5	0.3	128	72	62	68	71	72
0.5	0.4	159	109	86	97	100	102
0.5	0.5	190	132	112	128	130	131
0.7	0.1	93	31	27	28	30	31
0.7	0.2	155	73	64	72	73	77
0.7	0.3	234	126	105	115	116	118
0.7	0.4	292	173	149	161	165	167
0.7	0.5	355	229	198	219	224	229
0.9	0.1	252	70	47	49	53	55
0.9	0.2	392	136	111	115	117	117
0.9	0.3	490	234	180	193	196	199
0.9	0.4	578	323	251	278	289	291
0.9	0.5	658	394	335	376	378	381

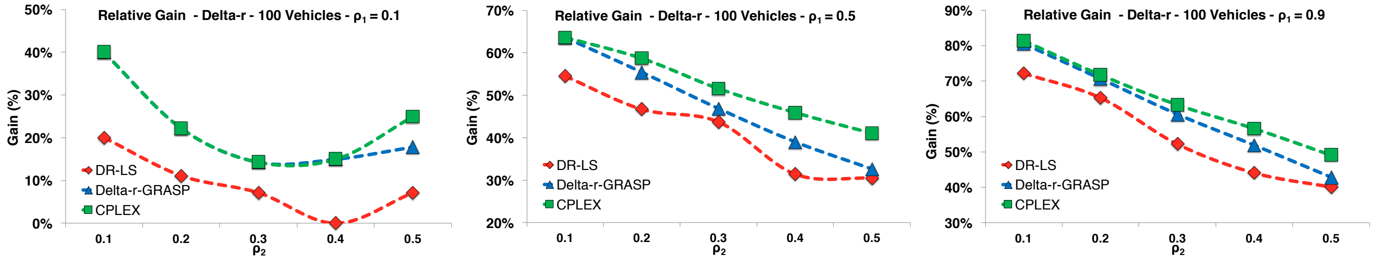
The Figure 7 presents the relative distance between the solution found by algorithm Delta-r-GRASP and the optimal value. As we can note in Figure 7, in all tested instances, our solution is less than 15% distance from the optimal one. Besides, in 6 out 25 tested instances, our algorithm finds the optimal value.

B. Solving the entire Cologne mobility trace

In the previous section, we compare our algorithm solution with Delta-r and with solver CPLEX, using an instance with 100 vehicles. As we can note in Table I our algorithm can find up to 80% fewer RSUs than Delta-r algorithm in order to achieve the same QoS. Besides, we show that for all tested instances, our solution differs no more than 15% from the optimal value. In this section, we will compare our algorithm with the entire Cologne mobility trace (composed of 75,515 vehicles). Our goal is to measure the influence of the Delta-r-GRASP algorithm for a bigger instance. As solver CPLEX is not able to find the optimal values for this instance, we will just compare the Delta-r-GRASP algorithm with the baseline Delta-r.

We also present the results for 25 different pairs of ρ_1 and ρ_2, and, for all instances, we run Delta-r-GRASP algorithm 11 times for four different α values. We use $\alpha = \{0.02, 0.05, 0.1, 0.2\}$ and present the results for the set with 11 solutions with the minimal number of RSUs. Due to the number of vehicles, we set the $Max\_Iterations$ parameter to 500.

Figure 8 presents the absolute number of RSUs saved by Delta-r-GRASP algorithm comparing with the baseline



(a) Relative Gain $\rho_1 = 0.1$

(b) Relative Gain $\rho_1 = 0.5$

(c) Relative Gain $\rho_1 = 0.9$

Figure 6: This Figure shows the gain, in percentage, between the number of RSUs computed by CPLEX (green dashed line), DR-LS (blue dashed line) and Delta-r-GRASP (red dashed line) algorithms for five different combinations of ρ_2 and three different ρ_1 using a 100 vehicles trace and comparing with Delta-r algorithm.

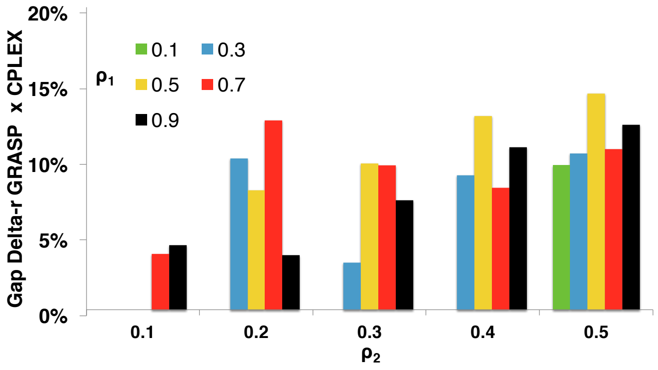


Figure 7: This Figure shows the distance, in percentage, between the number of RSUs computed by CPLEX (optimal value) and Delta-r-GRASP algorithms for different combinations of ρ_1 and ρ_2 using a 100 vehicles trace. The x-axis represents several values of ρ_2 while the y-axis represents the relative gain for distinct ρ_1 values when we compare CPLEX and Delta-r-GRASP algorithms. If our algorithm found the optimal value the respective bar does not appear.

algorithm Delta-r. In all tested instances, our approach finds less RSUs compared with Delta-r algorithm. For $\rho_1 = 0.9$ we saved up to 152 RSUs compared with Delta-r.

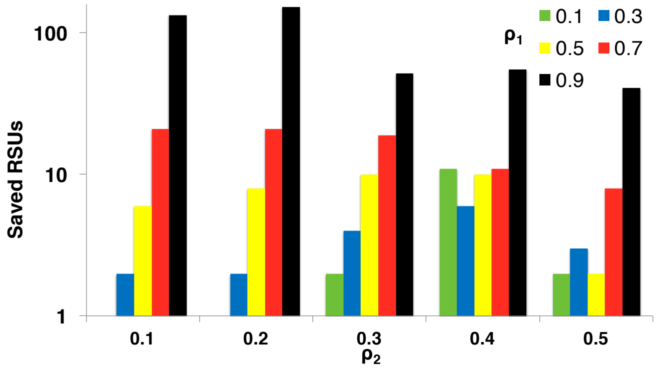


Figure 8: This Figure shows the absolute difference, in log scale, between the number of RSUs computed by Delta-r and Delta-r-GRASP algorithms for different combinations of ρ_1 and ρ_2 for the entire Cologne trace. The x-axis represents several values of ρ_2 while the y-axis represents the absolute gain for distinct ρ_1 values when we compare both algorithms.

Figure 9 presents the relative number of RSUs saved by Delta-r-GRASP algorithm comparing with the baseline algorithm Delta-r. For instance, when $\rho_1 = 0.9$ we saved up to 35% of RSUs compared with Delta-r algorithm solution.

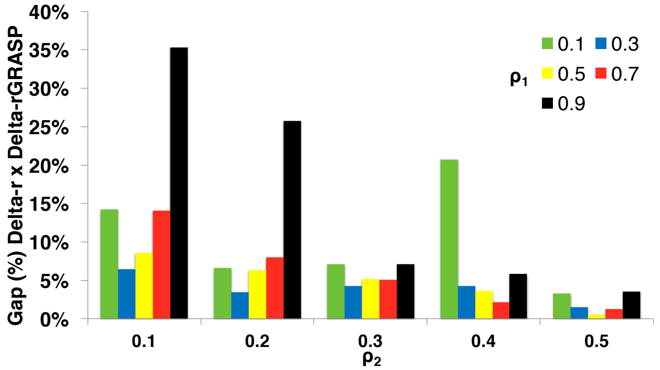


Figure 9: This Figure shows the relative difference, in percentage, between the number of RSUs computed by Delta-r and Delta-r-GRASP algorithms for different combinations of ρ_1 and ρ_2 for the entire Cologne trace. The x-axis represents several values of ρ_2 while the y-axis represents the absolute gain for distinct ρ_1 values when we compare both algorithms.

IX. Final Remarks

In this work, we proposed the Delta-r-GRASP algorithm for allocating the roadside infrastructure supporting vehicular networks using the Delta Deployment [3] metric. Although other works presented an Integer Linear Programming Formulation for this problem, this is the first work that presented exact results and compared them with a heuristic result. Besides, we showed that our results are differed no more than 15% from the optimal value for all tested instances. However, the main contribution of this work was the proposed algorithm (Delta-r-GRASP) that obtained results with gains up to 80% when compared with baseline Delta-r algorithm proposed by Sarubbi and Silva [15].

As a future work, we intend to present even better approximate algorithms using other local search neighborhoods and other techniques as Variable Neighborhood Search (VNS) or Variable Neighborhood Descend (VND) metaheuristics.

Acknowledgements

This work was partially funded by CNPq, FAPEMIG, CAPES and CEFETMG.

References

[1] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE*, vol. 46, pp. 164–171, June 2008.

[2] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, P. Manzoni, and M. Sanchez, "Prototyping an automatic notification scheme for traffic accidents in vehicular networks," in *4th IFIP Wireless Days (WD)*, pp. 1–5, 2011.

[3] C. M. Silva and W. Meira, "Evaluating the performance of heterogeneous vehicular networks," in *Vehicular Technology Conference (VTC), 2015 IEEE*, Sep 2015.

[4] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1609–1615, IEEE, 2011.

[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *ACM MobiSys*, 2008.

[6] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Optimal content downloading in vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, p. 1377–1391, 2013.

[7] S. Smaldone, L. Han, P. Shankar, and L. Iftode, "Roadspeak: enabling voice chat on roadways using vehicular social networks," in *Proceedings of the 1st Workshop on Social Network Systems*, pp. 43–48, ACM, 2008.

[8] J. Blum, A. Eskandarian, and L. Hoffman, "Challenges of intervehicle ad hoc networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, pp. 347–351, Dec 2004.

[9] K. Mershad, H. Artail, and M. Gerla, "Roamer: Roadside units as message routers in vanets," *Ad Hoc Networks*, vol. 10, no. 3, pp. 479 – 496, 2012.

[10] A. Reis, S. Sargento, and O. Tonguz, "On the performance of sparse vehicular networks with road side units," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1–5, May 2011.

[11] Y. Wu, Y. Zhu, and B. Li, "Infrastructure-assisted routing in vehicular networks," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1485–1493, IEEE, 2012.

[12] J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Road side unit deployment: A density-based approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, pp. 30–39, Jul 2013.

[13] A. Kchiche and F. Kamoun, "Centrality-based access-points deployment for vehicular networks," in *Telecommunications (ICT), 2010 IEEE 17th International Conference on*, pp. 700–706, IEEE, Doha 2010.

[14] T. Feo and M. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–135, 1995.

[15] J. F. M. Sarubbi and C. M. Silva, "Delta-r: A novel and more economic strategy for allocating the roadside infrastructure in vehicular networks with guaranteed levels of performance," in *IEEE/IFIP Network Operations and Management Symposium (NOMS), 2016 IEEE*, Apr 2016.

[16] Z. Zheng, P. Sinha, and S. Kumar, "Alpha coverage: Bounding the interconnection gap for vehicular internet access," in *INFOCOM 2009, IEEE*, pp. 2831–2835, April 2009.

[17] Z. Zheng, Z. Lu, P. Sinha, and S. Kumar, "Maximizing the contact opportunity for vehicular internet access," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, March 2010.

[18] J. Lee and C. Kim, "A roadside unit placement scheme for vehicular telematics networks," in *Advances in Computer Science and Information Technology* (T.-h. Kim and H. Adeli, eds.), vol. 6059 of *Lecture Notes in Computer Science*, pp. 196–202, Springer Berlin Heidelberg, 2010.

[19] O. Trullols, M. Fiore, C. Casetti, C. Chiasserini, and J. B. Ordinas, "Planning roadside infrastructure for information dissemination in intelligent transportation systems," *Computer Communications*, vol. 33, no. 4, pp. 432 – 442, 2010.

[20] R. Church and C. R. Velle, "The maximal covering location problem," *Papers in regional science*, vol. 32, no. 1, pp. 101–118, 1974.

[21] M. Nekoui, A. Eslami, and H. Pishro-Nik, "The capacity of vehicular ad hoc networks with infrastructure," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pp. 267–272, April 2008.

[22] J. Barrachina, J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "V2x-d: a vehicular density estimation system that combines v2v and v2i communications," in *IEEE/IFIP Wireless Days*, (Valencia, Spain), November 2013.

[23] G. Marfia, G. Pau, E. De Sena, E. Giordano, and M. Gerla, "Evaluating vehicle network strategies for downtown portland: opportunistic infrastructure and the importance of realistic mobility models," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pp. 47–51, ACM, 2007.

[24] O. Tonguz and W. Viriyasitavat, "Cars as roadside units: a self-organizing network solution," *Communications Magazine, IEEE*, vol. 51, pp. 112–120, December 2013.

[25] O. Trullols-Cruces, M. Fiore, and J. Barcelo-Ordinas, "Cooperative download in vehicular environments," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 4, pp. 663–678, 2012.

[26] B. Aslam, F. Amjad, and C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," in *IEEE Symposium in Computers and Communications (ISCC), 2012 IEEE*, (Cappadocia, Turkey), pp. 423 – 429, July 2012.

[27] J. F. M. Sarubbi, F. V. C. Martins, and C. M. Silva, "A genetic algorithm for deploying roadside units in vanets," in *IEEE Congress on Evolutionary Computation (CEC), 2016 IEEE*, jul 2016.

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 3rd ed., 2009.