An Optimized Roadside Units (RSU) Placement for Delay-Sensitive Applications in Vehicular Networks

Sara Mehar, Sidi-Mohammed Senouci, Ali Kies, Zoulikha Mekkakia Maaza

An Optimized Roadside Units (RSU) Placement for Delay-Sensitive Applications in Vehicular Networks

Sara Mehar[*1], Sidi Mohammed Senouci[*2],

[*] DRIVE Labs, University of Burgundy, Nevers, France
[1,2]{Sara.Mehar, Sidi-Mohammed.Senouci}@u-bourgogne.fr

Ali Kies[#3], MekkakiaMaazaZoulikha[#4]
[#]Department of Data processing
University of Sciences and the Technology of Oran, USTO, Oran, Algeria
[3,4]{kies_ali,mekkakia}@univ-usto.dz

Abstract— **Over the last few years, a lot of applications have been developed for Vehicular Ad Hoc NETworks (VANETs) to exchange information between vehicles. However, VANET is basically a Delay Tolerant Network (DTN) characterized by intermittent connectivity, long delays and message losses especially in low density regions [1]. Thus, VANET requires the use of an infrastructure such as Roadside Units (RSUs) that permits to enhance the network connectivity. Nevertheless, due to their deployment cost, RSUs need to be optimally deployed. Hence, the main objective of this work is to provide an optimized RSUs placement for delay-sensitive applications in vehicular networks that improves the end-to-end application delay and reduces the deployment cost. In this paper, we first mathematically model the placement problem as an optimization problem. Then, we propose our novel solution called ODEL. ODEL is a two-steps technique that places RSUs only in useful locations and allows both vehicle-to-vehicle and vehicle-to-infrastructure communication: (i) the first step is comprehensive study that looks for the RSUs candidates locations based on connectivity information, and (ii) the second step uses genetic algorithm and Dijkstra algorithm to reduce the number of RSUs based on the deliverance time requirement and the deployment cost. We show the effectiveness of our solution for different scenarios in terms of applications delay (reduced by up to 84%) and algorithm efficiency (computation performance reduced by up to 79% and deployment cost reduced at least by up to 23%).**

Keywords: Covering location optimization, real-time applications, delay constraints, genetic algorithm, deployment cost.

I. INTRODUCTION

In the latest years, the innovative progress in both transportation and communications technologies made information exchange possible on roads using smart cars [2]. Through communication, Vehicular Ad Hoc Network (VANET) is created. VANET is composed of a set of high mobile nodes, which results in a frequently changing network topology with intermittent connectivity. In VANET, there are two possible ways to communicate: vehicle to vehicle (V2V), vehicle to infrastructure (V2I) or both called V2X [3]. V2I communication capabilities have a huge impact on the network reliability and information exchange [2]. To enhance the network coverage and hence the quality of services, the infrastructure must be optimally installed. Infrastructures deployment is not trivial and needs intensive investigations. In this paper, we focus on road-side units (RSUs) placement. RSUs act as wireless access points. We studied and analyzed the existing works to find out their limitations. Some works

focus on urban scenarios and propose to put RSUs in intersections to communicate information to a large number of cars; some others suggest distributing RSUs randomly or uniformly on the road. In addition, most of related work limit the communication to the RSUs only instead of using both V2I and V2V communications. Consequently, initial RSUs locations are not optimized to support frequent network disconnection. Hence, they are not adapted to delay-sensitive applications as the delay constraint is not respected.

In this work, we propose a novel location solution called ODEL (An Optimized Roadside Units Placement for Delay-Sensitive Applications in Vehicular Networks). The later is a two-steps solution that optimizes RSUs deployment for vehicular networks applications with delay sensitive constraints. In the first step, we develop an algorithm to find the most relevant RSUs candidate locations based on network connectivity and hence avoiding the useless positions with high density where vehicles can communicate with each other's without using any additional infrastructure. In fact, from a given map and traffic information (density, speed), we calculate the probability of connectivity in each road segment. Then, we choose RSUs candidate positions in segments' midpoints with low connectivity probability. In segments with high probability of connectivity, the information is exchanged using V2V communication capabilities. The second step is an optimization phase in which we select the best RSU locations for final installation based on both the application constraints in terms of delay and the infrastructure deployment cost. For that, we combine two optimization algorithms: Genetic heuristic [4] and Dijkstra algorithm [5] as detailed in Section IV. The results show that ODEL allows reducing significantly both the communication delay and the deployment cost.

This paper is structured in five sections: In the second Section, we introduce the most important related work about Road-Side units placement. We give the mathematical modeling and the corresponding complexity of RSUs location in Section III. In Section IV, we describe the main steps of the resolution and evaluate the performances of our solution in Section V. Section VI summarizes and concludes the paper.

II. RELATED WORK

Road-side units positioning is a covering location optimization problem. In covering location problems, the goal is to find the optimal positions to cover all the clients (vehicles), while taking several constraints into account such as

deployment cost and applications quality of services requirements. A plethora of work is done to cover the field of RSUs optimized siting. In [6][7][8][9], the authors aim to place RSUs optimally in order to enhance connectivity. They consider intersections as best locations in urban scenarios since the density in intersections is usually higher and information can propagate in all directions. In [6], the authors aim to improve data dissemination in an urban area. They model the location problem as a Maximum coverage with Time Threshold Problem (MCTTP [10]). To optimally deploy RSUs to cover the maximum vehicles in the road, they use a genetic heuristic to solve the MCTTP and get the best RSUs placement. In [7], the goal is to find the optimal positions for RSUs to cover the vehicles in urban roads and reduce the delay of safety messages propagation. The authors consider intersection positions as potential locations to deploy RSUs. They mathematically model the problem and use two methods to resolve it: (i) analytical Binary Integer Programming (BIP) [11] to find best time to broadcast in the whole area, and (ii) Balloon Expansion Heuristic (BEH) method to find best time over each route. In [8], the authors propose a cost-efficient RSUs deployment scheme. They aim to update security certificates in urban area within short delay using the deployed RSUs. In this scheme, they suppose that each vehicle can communicate with RSUs in bounded driving time whatever its actual position, and the extra overhead time used for adjusting routes to update certificates. They model the problem as a set-covering problem and use the polynomial-time approximation algorithm called "Greed Set Cover" [12] to solve it. In [9], the authors propose an RSUs siting strategy for file downloading in the case of urban scenario. The main goal is to guarantee file downloading success ratio and delay requirement with the lowest deployment cost. They also use intersections as RSUs initial placement and consider only V2I communications. To improve file downloading success ratio, they first model the inter-meeting time between vehicles and RSUs as a time continuous homogeneous Markov chain with two states (disconnected and connected to RSU). Then, they deduce the objective function that relates RSUs deployment and file download ratio success and delay. After that, they model the road network as a weighted undirected graph where each edge represents the average passing time on the corresponding road. Finally, they use the depth-first traversal algorithm to get optimal placement to deploy RSUs. However, the authors in [6][7][8][9] consider only V2I communications, which introduce a large delay when no car is available to drive toward the RSUs. In [13], the authors study the problem of deploying gateways to provide the required coverage while minimizing the deployment cost in urban area. They formulate the problem as a Maximum Coverage Problem (MCP) [14].They extract and analyze traffic data over several hours in a given area. They divide the studied area in uniform zones and count the number of vehicles entering and leaving each zone for each time unit to obtain the distribution of transition probability between two zones. The later is used to find the candidate zones to deploy gateways. Finally, they develop their own heuristic algorithm called "MobGDeploy" to find the optimal gateways locations. However, the authors try to cover the maximum clients by choosing to put gateways in the zones with high density; so the solution does not take into account the

delay requirement especially in low-density cases. Such a solution is also not efficient to transmit safety messages for example. In [15], the authors aim to optimize the travel time between neighbors landmarks. They try to optimally find the best RSUs positions in urban area in order to efficiently aggregate traffic information on the road and use it to improve route planning. They use genetic algorithm to find best positions. For each iteration of the genetic algorithm, they use NS2 [16] simulator to evaluate the current solutions and to estimate the required travel time by simulation. However, they choose a random candidates deployment positions, which increase the complexity of the solution and does not take into account density information. In [17], the authors' goal is to study the size of the gaps between RSUs to improve data collection and delivery ratio in a highway scenario. They initially distribute RSUs along the road equidistantly and formulate the distance upper bound between RSUs. Then, they use Omnet++ simulator [18] to evaluate the considered system under realistic settings to find the best gap between every two successive RSUs. In [19], the authors' goal is to minimize the average reporting time of information to a given RSU. To effectively collect data in highway scenario, they propose to use a uniform initial RSUs distribution (each twice radio range distance). To reduce the deployment cost and get the best RSUs placement, they develop a heuristic based on balloon expansion method. However, the authors in [17] and [19] propose their solution only for a single road. In addition, the uniform distribution is not the best initial distribution, as it does not take into account the density of the cars in the road and increase the delivery delay of information. In [20], the authors model the network connectivity using a fluid model and a stochastic model. They propose RSUs assignment as an application of their model. The fluid model is used to compute the network density. The stochastic model takes into account the vehicles' random behavior. This model is different than the Poisson-arrival-location-model (PALM) described in [21]. In fact, this model is designed for urban scenarios; it takes into account traffic light and interactions between vehicles. It allows determining the degree of connectivity in a given road. Then, it assigns the RSU to enhance the connectivity where the number of connected nodes is lower than a threshold. However, the authors find the best positions to place RSUs although they do not optimize the number of RSUs. In [22] the authors study the connectivity in VANETs based on results of percolation theory. They consider cars density, the rate of equipped vehicles, and communication coverage to study the distribution of isolated nodes and the impact of putting the RSU in intersections. The authors show that RSUs placement in crossroads does not impact significantly the proportion of isolated cars. From the related work, we can observe that the problem of RSUs deployment depends on the application. Many works consider only V2I communication i.e. the vehicle must be in the RSUs' radio range to communicate messages. In addition, the authors choose to use intersections in the case of urban scenario and uniform distribution in the case of highway scenario, which are not the best RSUs positions. Consequently, in our work, we deal with RSUs deployment in both urban and highway scenarios for delay-sensitive applications. We use V2X communications to enhance the network connectivity and reduce unnecessary infrastructures.

III. Optimization Problem Modeling

For our siting problem, we aim to find the best RSUs locations in urban or highway scenarios. The challenges consist on ensuring the application delay constraint and reducing the deployment cost. In fact, our objective is to find the best RSUs placement and respect the delay required to transmit messages while minimizing the number of RSUs. In this work, we take into account the following assumptions: we consider V2X communications i.e. messages are transmitted through both vehicles and RSUs; we assume that all RSUs are connected to each other through wired links (therefore, we considered only the message delivery delay to the closest RSU. The delay from this RSU to the final destination – a server within the cloud – is supposed fixed); for initial distribution positions, we consider low connectivity segments as the best RSUs location candidates, so our solution can be applied to both urban and highway scenarios; then, for final placement, we reduce the delay constraint and deployment cost by modeling the location problem and solving it. Herein we give our mathematical modeling of RSUs siting problem followed by the complexity details of the exact solution.

A. Optimization problem modeling

Let I={1,…,m} be the set of candidate positions to install RSUs, a_i be the RSUi deployment cost, J={1,…,n} be the set of all road segments in the studied area and t_j be the required delay to route the message from road segment j to the closest RSU. Let also y_i be the variable that indicates if the RSU I is open (deployed) or not. These variables are summarized in Table I.

TABLE I. NOTATION

Variable	Notation
deployment cost to build RSU i	a_i
Decision variable to open or not the RSU$_i$	yi
Minimum delay to transmit a message from a vehicle in road segment j to the closest RSU	t_j
The number of RSU candidates	m
The number of road segments	n
Vehicles' radio range	r
Application delay threshold	$T\_threshold$

To solve the RSU placement problem, we model it using equations (1) to (6). Equation (1) represents the objective function to minimize. It is a weighted function composed of two sub-functions: (i) *F1* representing the RSUs installation cost and (ii) *F2* representing the delay to reach the closest RSU.

$$F = \alpha F1 + \beta F2 \qquad (1)$$

Where α and β represent the weight of each sub-function. We choose α=1/3, β=2/3 as the delay constraint is the most important.

$$F1 = min \sum_{i=1}^{m} a_i y_i \qquad (2)$$

$$F2 = min \sum_{j=1}^{n} t_j \qquad (3)$$

$$t_j = \min_{1 < s < m} t_{js} \qquad (4)$$

Equation (4) indicates the shortest delay t_j to send messages from road segment j to any RSU and t_{js} is the required delay to send a message from segment j to the RSU number s. The solution feasibility must respect the constraints given by equations: (5) to ensure that delay t_j to transmit message from road segment j is lower than a given application threshold $T\_threshold$; (6) and (7) represent the integrity constraints.

$$t_j < T_{threshold} \ \forall j \in \{1 \dots n\} \qquad (5)$$

$$\sum_{i=1}^{m} y_i \le m \qquad (6)$$

$$y_i \in \{0,1\} \qquad (7)$$

B. Exhaustive solution complexity

RSUs placement is a NP-hard problem [23] and requires huge computation capacities to find the best solution. We give in equation (8) the complexity of exhaustive solution to show the number of possible combinations to solve it. In fact, in this problem, we need to find the optimal number of RSUs to be deployed and their optimal positions. Therefore, we need to find the optimal number of k RSUs from m possible initial positions (see table I) ($k<=m$). We can write it as a mathematic combination C_m^k. However, k is unknown and it is in the interval $\{1...m\}$. Consequently, the problem complexity becomes $\sum_{k=1}^{m} C_m^k$. In addition, to calculate the time to the closest RSU, we need to apply a short path algorithm to evaluate the delay to send messages to nearest RSUs for each solution combination. Therefore, for each possible solution, we add the complexity of the short path algorithm that is given by $o(n^2)$, where n is the number of total segments on the studied area (see table I). We get finally equation (8).

$$\sum_{k=1}^{m} C_m^k * o(n^2) \qquad (8)$$

Due to the problem complexity and the heavy computation required by the exhaustive method, we propose to use a heuristic to solve it in order to reduce the resolution computation time and get the best solution. In the next section, we explain in details our proposed solution ODEL and the several adaptations introduced to the genetic heuristic to solve the delay-sensitive RSUs placement problem.

IV. ODEL Steps

In our work, the objective is to find the optimal number of RSUs to be deployed and their optimal positions while respecting the application delay requirements and the

deployment cost constraint. To do that, we propose two processing steps as depicted in Fig. 1.
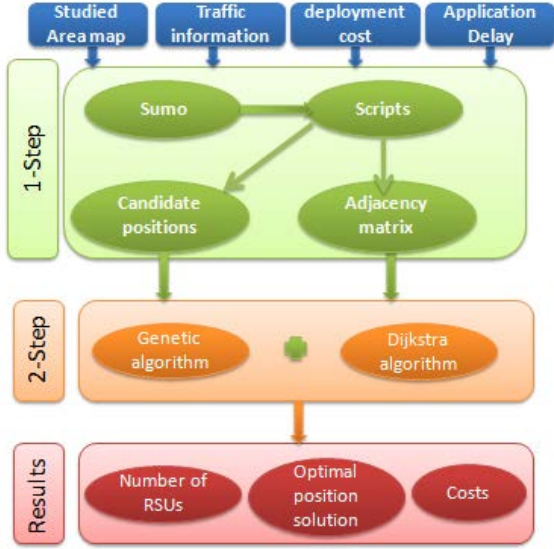

Figure 1. ODEL solution

To apply ODEL, we need to introduce several parameters such as: the map of the studied area, the traffic information, the application delay constraints and deployment cost information. Then, we apply the two steps solution as explained below.

A. First-step resolution: initial selection of RSUs candidate positions

In this phase, we calculate RSUs initial positions. In contrast to related work that choose initial positions randomly, uniformly or in intersections, we study the area and find the strategic positions based on the probability of connectivity. In fact, several work use the connectivity to optimize the information collection and dissemination between vehicles or mobile nodes [24][25][26][27][28]. Though, the connectivity information is not used to determine best RSU positions except in [20]. However, the authors did not optimize the number of RSUs candidates. Herein we explain how to get the initial candidates positions. For a given area, we extract information about density and speed in each segment of length 2*r meters using Simulation of Urban Mobility (SUMO) [29]. Then, we use some scripts that we developed to calculate the connectivity probability. For urban scenario we use formula of connectivity probability (9) and (10) proposed in [22]. Due to lack of space, the formula for a highway scenario is not described in this paper; but the reader can refer to [30] for more details.

$$pu = \begin{cases} 1 & if\ 0 \le L < r \\ \sum_{i=0}^{\lfloor L/r \rfloor} \dfrac{\left(-\lambda e^{-\lambda r}(L - ir)\right)^{i}}{i!} \\ \quad - e^{-\lambda r} \sum_{i=0}^{\lfloor L/r \rfloor - 1} \dfrac{\left(-\lambda e^{-\lambda r}(L - (i+1)r)\right)^{i}}{i!} & if\ L \ge r \end{cases} \quad (9)$$

$$pl = \sum_{i=0}^{\lfloor L/r \rfloor + 1} \dfrac{\left(-\lambda e^{-\lambda r}(L - (i-1)r)\right)^{i}}{i!}$$

$$\quad - e^{-\lambda r} \sum_{i=0}^{\lfloor L/r \rfloor} \dfrac{\left(-\lambda e^{-\lambda r}(L - ir)\right)^{i}}{i!} \quad (10)$$

Where *pu* is the probability that there is a sequence of connected nodes in the road segment, *pl* is the sufficient condition to have a distance between two consecutive vehicles at most equal to the radio range r. λ is the vehicle arrival rate and L is the road segment length. After that, we choose the segment with lower connectivity probability as best candidate locations.

Also, based on the traffic information and road topology, we deduce the adjacency matrix. This matrix contains the neighborhoods of all road segments and the propagation delay in each road segment.

B. Second-step resolution: Optimal placement

As mentioned above, our solution takes into consideration both the application delay requirement and the deployment cost related to RSU deployment. To ensure that, we implement two optimization algorithms: (i) Dijkstra to find best routing time from any segment on the road to the closest RSU, and (ii) Genetic algorithm to find the best deployment positions for RSUs that satisfies the application time constraints and takes into account the deployment cost. This cost includes RSUs and network installation costs. The main entries of the second-step are: the adjacency matrix (a neighborhood matrix), the RSUs candidates vector and related deployment cost. We explain below our optimization algorithm with the different adaptations introduced to resolve RSUs locations problem.

1) Genetic algorithm

We apply the Genetic algorithm illustrated in Algorithm 1. We run this algorithm many times. Each iteration consists of natural operations: selection, crossover, mutation and recombination [4]. In order to enhance the efficiency of the genetic algorithm, we use the Gaussian mutation [31] to create a new chromosome by modifying one or more genes value and Roulette-wheel selection [32] to stochastically choose new offspring chromosomes from the pool for the new generation.

Algorithm 1 Genetic algorithm

<Adjacency matrix Madj, RSU Candidate Tcand, deployment cost>

1. Pop.Create ("random", initial population)
2. Pop.CalculateFitness()
3. **While** (!Max _Iterations)
4. {Offspring = Select ("Roulette- wheel", Pop, Nb-chromosoms)
5. Offspring.Crossover(uniform, P_c) ;
6. Offspring.Mutation("Gaussian", P_m) ;
7. Offspring.Permut () ;
8. Recombination("Keep-Best", Offspring, Pop)
9. } Pop.BestSolution() <Nb CS, position CS>

Where P_c is the crossover probability and P_m is the mutation probability. The genetic algorithm generates and evaluates many solution combinations coded as a genetic chromosome or code and look for the best one as final results.

2) Genetic code

We code each solution combination using a suitable binary code as depicted in Fig. 2. The binary code is composed of m bits; each bit is equal to one if the RSU is on, else it takes the value 0. We evaluate the chromosome using the objective function detailed below.
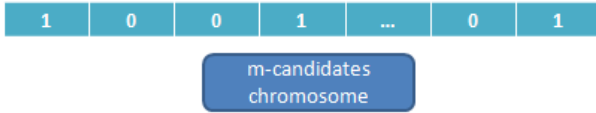

Figure 2.   Binary-chromosome

3) Objective function

Equation (1) is equivalent to equation (12). This formula is used in the genetic algorithm to evaluate each chromosome. In this equation, we calculate the deployment cost and the delay required to transmit messages to the closest RSU using our adapted Dijkstra algorithm as further explained below.

$$F = \min(\alpha \sum_{i=1}^{m} a_{i+}$$

$$\beta \sum_{j=1}^{n} \min_{i=1\dots m}(Dijkstra(j,i))) \tag{12}$$

4) Dijkstra algorithm adaptation

We adapt the Dijkstra algorithm to our problem to reduce its complexity from o(n²) to o($Nb*n$). In fact, each vertex has at most Nb neighbors. Given the adjacency matrix "$Madj$" got from the first-step of the solution, we construct a graph "G" to apply Dijkstra algorithm, each edges E in the graph G is the distance in term of delay from road segment $v1$ to its neighbor $v2$. Dijkstra calculates the delay from a given source vertex to all other vertices in the graph G. Only the shortest path toward the RSUs candidates is used to evaluate the solution. The Dijkstra adaptation is mainly done in line 9 of Algorithm 2.

Algorithm2 Dijkstra algorithm
<Adjacency matrix Madj, RSU Candidate Tcand , Source vertex s>

1.   *// Initialization*
2.   *for i = 0 to |V| - 1*
3.   *Tdistance [i] = INFINITY*
4.   *Previous [i] = NULL*
5.   *end*
6.   *dist[s] =0;*
7.   *while(List.size() !=Nb_ vertex)*
8.   *pick a closest vertex v from Unvisited and add it to List*
9.   *for each edge of v find the closest neighbor v2*
10. *if (Tdistance [v1] + length(v1, v2) <Tdistance [v2])*
11. *Tdistance [v2] = Tdistance [v1] + length(v1, v2)*
12. *Previous[v2] = v1*

---

13. *end if*
14. *end for*
15. *end while*

Where "*Tdistance*" is an array of distances from the source "*s*" to each vertex *V*. "*Previous*" is an array of vertices' predecessors, "*List*" is the list of visited vertices and "*Unvisited*" is a set of remaining vertices.

In the next Section we present and analyze the different ODEL's results.

V.   RESULTS AND PERFORMANCE

We develop automatic scripts in order to generate different maps, traffic scenarios and to extract density and speed information in each road segment using traffic simulator [29], awk, shell and xml. The density changes over the time and from one road segment to another. Then, we develop Dijkstra and genetic algorithms and the different adaptations inside evolutionary framework proposed in [33] using c++. We run several tests with different scenarios for both low and high density. We carry out all the performance tests on virtual machine with Intel i5 64Bits processor and 3 Gb. of memory. In Table II we present the parameters used in our tests and we summarize the optimization results in Table III, IV and V.

TABLE II.   PARAMETERS TABLE

Parameter	Value
Radio range r	250
Number of total road segments	60-500
Number of RSU candidates	25-150
Road density (vehicle/km)	1-30
Vehicles average speed (km/h)	15-50
Road segment length (m)	500
T_threshold (s)	0.1-10
Crossover probability Pc	0.3-0.5
Permutation probability Pm	0.8
Population size	50

To evaluate our solution's performances, we choose three criteria: (i) the delivery delay required to send messages from any road segment toward the nearest RSUs, (ii) the number of necessary RSUs to guarantee applications delay requirements and (iii) the computation performance in term of number of iterations to find best solutions. In RSUs siting problem, we want to activate only some RSUs candidates. In fact we do not need to cover the whole area. Table III illustrates 9 different examples; for each example we give the total number of segments, the number of possible candidates positions found in step 1, and the number of RSUs found as best solution in step 2. The cost efficiency is obtained by dividing the number of RSUs "off" (non-deployed) resulting from step 2 by the number of RSUs candidates resulting from step 1. These results show that our solution can meet the requirement of the application without siting unnecessary RSUs. We note that the optimization results depend on the delay constraint *T_threshold*

of the application given in the last column of table V (e.g. in example 1, the total number of segments in the studied area is 60, it requires 15 RSUs candidates by applying step 1 and 11 RSUs "On" (deployed) by applying step 2 to ensure the application threshold of 3 seconds is satisfied. In example 2 (respectively 3), the number of RSUs "On" (deployed) after the step 2 is 26 (respectively 7) to have a delay lower than the T_threshold of 1 second (respectively 5 seconds.)).The cost efficiency results show that our solution reduces at least 23% of the deployment cost comparing to the deployment cost without the optimization of step 2. And it could reach 79% of efficiency if the delay constraint are larger (see example 3).

TABLE III.   RESULTS1: FINAL RSU NUMBER

Example	Total number of segments	Application delay T_threshold	Candidate segments number (found in step1)	Number of final positions (found in step2)	Cost Efficiency
1.	60	3	15	11	26.66%
2.	60	1	34	26	23.52%
3.	60	5	34	7	79.41%
4.	144	5	25	19	24%
5.	144	5	64	42	34%
6.	144	7	64	33	48%
7.	220	5	60	40	33.33%
8.	220	6.5	88	37	57.95%
9.	220	7	74	26	64.86%

For the same examples of Table III, we compare in Table IV between the number of iterations (complexity) of our Genetic algorithm and those required by the exhaustive solution explained further above in equation (8). Equation (8) is equivalent to 2^m by using the generalized hypergeometric function [34]. The number of iterations depends on the number of total road segments and the number of candidate positions as defined in table III. We can see in table IV that the number of iterations in our solution is significantly lower. We note that the number of iterations increases at expense of the total number of segments in the studied examples as the delay is measured in each road segment and not only in candidate ones. However, it is always lower than the number of iterations of the exhaustive resolution method (e.g. In the example 1 in table IV, genetic solution requires 31 times unit while the exhaustive solution for the same scenario requires 32768 times unit, which corresponds to 2^{15} combination in exact solution).

TABLE IV.   RESULTS2: COMPUTATION

Example	Number of exhaustive iteration	Number of genetic iteration	Application delay T_threshold	Genetic Computation time
1.	32768	1000	3	31
2.	$\cong 1717\times10^{7}$	1000	1	30.39
3.	$\cong 1717\times10^{7}$	1000	5	19.6
4.	33×10^{6}	1000	5	141
5.	$\cong 18\times10^{18}$	1000	5	105
6.	$\cong 18\times10^{18}$	1000	7	120
7.	$\cong 1.15\times10^{18}$	1000	5	1100
8.	$\cong 3.09\times10^{26}$	1000	6.5	1500
9.	$\cong 1.8\times10^{22}$	1000	7	532.31

In Table V, we calculate the delay efficiency of our solution; we compare the minimum delay to deliver messages from all road segments to the closest RSUs. These results show that our solution is a much more effective (e.g. for example 1, the average delay required to transmit data to the destination is 0.31 second which is lower than 340 second without using RSU but only V2V communications).

TABLE V.   .RESULTS3: DELAY EFFICIENCY

Example	Delay when no RSU is deployed	Delay with optimal RSU number	Average Delay	Delay Efficiency	Application delay T_threshold
1.	340.4	18.76	0.31	94.5%	3
2.	423.74	11.73	0.19	97.2%	1
3.	340.4	53.23	0.88	84.36%	5
4.	1279.22	388.93	2.7	69.66%	5
5.	1762.4	105	0.72	94.04%	5
6.	1762.4	157.74	1.09	89.5%	7
7.	1359.06	99.36	0.45	92.68%	5
8.	1436.83	99.36	0.45	93.08%	6.5
9.	1019.26	156.65	0.71	84.63	7

Finally, simulation results show that our solution ODEL achieves desirable efficiency in term of delay requirement, while preserving necessary coverage in the whole area using both V2I and V2V communication capabilities. In addition, our solution is designed to support large scale scenarios and reduce significantly the computation operation. We note that ODEL aims to improve the delay by deploying RSUs in road with low density unlike the related work that improve connectivity only in the intersections. However, when the density is lower than the density considered in our study, neither our solution, nor the related work contributions achieve a low delay. In fact, our solution optimizes the number of deployed RSUs to ensure a full covered area over V2X communication. Moreover, when density is very low, it is more interesting to use 3G or LTE solutions as the number of messages is very low and does not consume huge cellular network resources.

VI. CONCLUSION AND FUTURE WORK

In this work, we focus on the delay requirement for delay-sensitive vehicular networks applications. We aim to reduce

the message communication delay and cope with intermittent connectivity by optimally siting new infrastructure (roadside units). To do that, we propose a mathematical modeling of infrastructure placement with high delay constraints and we propose ODEL solution. ODEL finds first strategic locations candidates based on a study of traffic information in the studied area. Then, it optimizes the number of final RSUs locations using two optimization algorithms Genetic and Dijkstra algorithms. In contrast of the existing work that deploy RSUs without considering application delay constraints and generally place RSUs randomly, uniformly or in intersections, ODEL provides very interesting delay boundaries and could be applied in different scenarios. As a future work, we plan to test our solution to optimize the routing and dissemination end-to-end delay for delay-sensitive applications and extend our solution to support different road geometry and radio obstructions.

REFERENCES

[1] M. Seligman, K. Fall, and P. Mundur, "Storage routing for DTN congestion control," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 10, pp. 1183–1196, Dec. 2007.

[2] lorenzo galati Giordano and L. Reggiani, *Vehicular Technologies - Deployment and Applications*. InTech, 2013.

[3] M. Almeida, *ADVANCES IN VEHICULAR NETWORKING*. .

[4] M. Maric, "An Efficient Genetic Algorithm for Solving the Multi-Level Uncapacitated Facility Location Problem.," *Comput. Informatics*, vol. 29, no. 2, pp. 183–201, 2010.

[5] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[6] E. S. Cavalcante, A. L. L. Aquino, G. L. Pappa, and A. A. F. Loureiro, "Roadside unit deployment for information dissemination in a VANET," in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12*, 2012, p. 27.

[7] B. Aslam, F. Amjad, and C. C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, 2012, pp. 000423–000429.

[8] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "Roadside Units Deployment for Efficient Short-Time Certificate Updating in VANETs," in *Communications (ICC), 2010 IEEE International Conference on*, 2010, no. January 2014, pp. 1–5.

[9] Y. Liu, J. Niu, J. Ma, and W. Wang, "File downloading oriented Roadside Units deployment for vehicular networks," *J. Syst. Archit.*, vol. 59, no. 10, pp. 938–946, Nov. 2013.

[10] O. Trullols, J. M. Barcelo-Ordinas, M. Fiore, C. Casetti, and C.-F. Chiasserini, "A Max Coverage Formulation for Information Dissemination in Vehicular Networks," in *2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2009, pp. 154–160.

[11] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks," *Comput. Networks*, vol. 52, no. 12, pp. 2419–2431, 2008.

[12] A. Aloisio, V. Izzo, and S. Rampone, "FPGA implementation of a greedy algorithm for set covering," in *14th IEEE-NPSS Real Time Conference, 2005.*, 2005, p. 5 pp.

[13] Y. Xiong, J. Ma, W. Wang, and J. Niu, "Optimal Roadside Gateway Deployment for VANETs," *Prz. ELEKTROTECHNICZNY*, no. 7, pp. 273–276, 2012.

[14] N. Apollonio and B. Simeone, "The maximum vertex coverage problem on bipartite graphs," *Discret. Appl. Math.*, vol. 165, no. 0, pp. 37–48, 2014.

[15] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, "Data aggregation and roadside unit placement for a vanet traffic information system," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, 2008, pp. 58–65.

[16] "The Network Simulator - ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/.

[17] M. Rashidi, I. Batros, T. K. Madsen, M. T. Riaz, and T. Paulin, "Placement of Road Side Units for floating car data collection in highway scenario," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, 2012, pp. 114–118.

[18] "Omnet++." [Online]. Available: http://www.omnetpp.org/.

[19] B. Aslam and C. C. Zou, "Optimal roadside units placement along highways," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, 2011, pp. 814–815.

[20] I. W.-H. Ho, K. K. Leung, and J. W. Polak, "Stochastic Model and Connectivity Dynamics for VANETs in Signalized Road Systems," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 195–208, Feb. 2011.

[21] W. a. Massey and W. Whitt, "Networks of infinite-server queues with nonstationary Poisson input," *Queueing Syst.*, vol. 13, no. 1–3, pp. 183–250, Mar. 1993.

[22] M. Kafsi, P. Papadimitratos, O. Dousse, T. Alpcan, and J. Hubaux, "VANET Connectivity Analysis," *IEEE AutoNet, New Orleans, LA, USA, December 2008*, 2008.

[23] S. Arora, D. Karger, and M. Karpinski, "Polynomial time approximation schemes for dense instances of NP -hard problems," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing - STOC 95*, 1995, pp. 284–293.

[24] L. T. Dung and B. An, "A modeling framework for supporting and evaluating performance of multi-hop paths in mobile ad-hoc wireless networks," *Comput. Math. with Appl.*, vol. 64, no. 5, pp. 1197–1205, Sep. 2012.

[25] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," in *IEEE Wireless Communications*, 2006, vol. 13, no. 5, pp. 52–57.

[26] K. Abboud and W. Zhuang, "Modeling and Analysis for Emergency Messaging Delay in Vehicular Ad Hoc Networks," *GLOBECOM 2009 - 2009 IEEE Glob. Telecommun. Conf.*, pp. 1–6, Nov. 2009.

[27] T. D. C. Little and A. Agarwal, "An information propagation scheme for VANETs," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, 2005, pp. 155–160.

[28] A. Agarwal, D. Starobinski, and T. D. C. Little, "Analytical Model for Message Propagation in Delay Tolerant Vehicular Ad Hoc Networks," in *VTC Spring 2008 - IEEE Vehicular Technology Conference*, 2008, no. May, pp. 3067–3071.

[29] "Sumo: Simulation of Urban MObility." [Online]. Available: http://sumo-sim.org/.

[30] M. M. Artimy, W. Robertson, and W. J. Phillips, "Connectivity in inter-vehicle ad hoc networks," in *Electrical and Computer Engineering, 2004. Canadian Conference on*, 2004, vol. 1, pp. 293–298 Vol.1.

[31] R. Tinos and S. Yang, "Evolution Strategies with q-Gaussian Mutation for Dynamic Optimization Problems," in *Neural Networks (SBRN), 2010 Eleventh Brazilian Symposium on*, 2010, pp. 223–228.

[32] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Phys. A Stat. Mech. its Appl.*, vol. 391, no. 6, pp. 2193–2196, 2012.

[33] Q. H. Nguyen, Y.-S. Ong, and M.-H. Lim, "A Probabilistic Memetic Framework," *Evol. Comput. IEEE Trans.*, vol. 13, no. 3, pp. 604–623, Jun. 2009.

[34] J. Dziok and H. M. Srivastava, "Classes of analytic functions associated with the generalized hypergeometric function," *Appl. Math. Comput.*, vol. 103, no. 1, pp. 1–13, 1999.