Fog Node Optimum Placement and Configuration Technique for VANETs

Rehab Shahin*, Ali A. El-Moursy†, Sherif M. Saif‡, Hazem M. Abbas* and Salwa M. Nassar‡
*Department of Computer and Systems Engineering,
Faculty of Engineering, Ain Shams University, Cairo, Egypt
Email: {rehab.abdellatif, hazem.abbas}@eng.asu.edu.eg
†Computer Engineering Department,
University of Sharjah, Sharjah, UAE
Email: aelmoursy@sharjah.ac.ae
‡Computers and Systems Department,
Electronics Research Institute, Cairo, Egypt
Email: {sherif_saif, salwa}@eri.sci.eg

***Abstract*—Intelligent Transportation Systems (ITS) are nowadays considered very important applications of smart cities. One of the most important technologies that are utilized to support ITS is Vehicular Ad-hoc Networks (VANETs). In VANETs, vehicles communicate with each other (V2V) or with the infrastructure (Roadside Units) (V2I). Roadside Units (RSUs) collect data from vehicles in the coverage area and send it to cloud servers through the Internet. Cloud servers have high performance computational and storage capabilities that ITS applications require for data processing. However, due to the real-time requirements of the ITS applications, cloud approach alone cannot be guaranteed to satisfy the strict time constraints due to long latency access of the centralized cloud server. Fog Computing is an emerging approach that extends the services of cloud computing to the edge of the network. Fog Computing can be utilized in VANETs through deployment of fog nodes into RSUs. One of the major challenges is identifying the optimum number, locations and computational capabilities of the RSUs particularly in urban regions where obstacles exist heavily inside the coverage area of the RSUs. In this paper, we consider the optimization problem of fog-based RSU placement where the objective is to maximize the achieved level of service quality in a cost-effective way. The problem is formulated as a Satisfiability Modulo Theories (SMT) problem and solved using Microsoft Z3. The proposed approach is able to generate a set of solutions as Pareto front. We obtained data from OpenStreetMap for Cairo city. Our approach outperforms other solutions in the literature in terms of cost.**

***Index Terms*—Fog Computing, VANET, Roadside Units**

I. Introduction

An important part in the planning process of smart cities is considering an efficient transportation system to cope with the increasing number of vehicles and maximize road safety. Vehicular Ad-hoc Network (VANET) in smart cities provides many services including autonomous driving and congestion avoidance applications. In VANET, a vehicle exchanges messages with several entities including other vehicles, traffic lights, road side units or even pedestrians. Road side units (RSUs) are considered the backbone of the VANET. RSUs collect data from vehicles in the coverage area and deliver them to the cloud for heavy data processing and the offline data analysis and statistics. They exchange messages using a wireless communication technology such as the Dedicated Short Range Communication (DSRC) technology [1], [2].

978-1-7281-6535-6/20/$31.00 ©2020 IEEE

Fog Computing is a promising emerging approach that extends the computation, storage and networking services of cloud computing to the edge of the network near the source of the generated data [3]. A Fog Computing architecture is a three layer architecture (the end device (data source), the fog layer, and the cloud layer). The fog layer is the intermediate layer between the data source and the cloud server. A fog node is a compute/storage node that is able to perform partial processing of time critical data while further processing is handled in the cloud layer. Examples of fog devices include end user devices, access points and routers. As Fog Computing is implemented at the edge of the network, it provides low latency access due to its location awareness. It improves QoS for streaming and real time applications [3]. Researchers proposed different solutions, applications and platforms based on the Fog Computing paradigm [4].

In the context of vehicular technologies, the concept of Fog Computing is adopted in many VANET applications. It is assumed that the fog nodes are deployed at the RSUs. The distribution and location of fog-based RSUs in a VANET have a direct impact on the achieved QoS of the application. Several optimization techniques have been applied to solve the RSU placement problem [5], [6]. Most of the work done aims to maximize the achieved QoS in terms of connectivity. The limited computational capabilities of the RSUs as fog nodes are not taken into consideration. Furthermore, the impact of obstacles in the coverage region is not considered in most of the proposed work. Obstacles have direct impact on signal propagation that causes signal attenuation and accordingly some areas turn to be out of coverage.

In this paper we develop Fog-based RSU Optimum Configuration and Localization (Fog-ROCL) technique to solve the optimization problem of RSU placement. We address the challenges of optimal fog-based RSU distribution in a target coverage area using a cost-effective approach to satisfy a set of quality measures including coverage and processing demand. The contributions of this paper are summarized as follows. We develop a new model for the RSU optimum distribution problem and formulate this problem as a Satisfiability Modulo Theories (SMT) problem. The proposed model features the heterogeneity of RSUs in terms of computational capabilities to add flexibility and reality to the fog system. Our proposed approach solves the optimization problem using Z3, the SMT solver, and generates the optimum trade-off between cost and the achieved QoS of each configuration.

The rest of the paper is organized as follows: in section II, the previous work in literature is reviewed. In section III, the Fog-ROCL method is explained. The experimental setup and methodology is described in section IV. Results are discussed in section V. Section VI includes conclusion and discussion.

II. Related Work

The optimization problem of RSU placement has been investigated in several prior art [5]–[7]. In [5], the authors propose a model that facilitates V2V and multi-hop communication. The authors propose a method based on genetic algorithm to solve the optimization problem. The objective is to find the number and optimal locations of RSUs while considering the application delay requirements. However, their model does not support the Fog Computing concept where the application response time and the QoS could be violated in a multi-hop communication network. They assume a fixed radius of coverage for the RSUs which is unrealistic since in obstacle-dense urban regions, there is high probability that signals hit an obstacle before reaching their destinations. This causes variable attenuation to the signals that depends on the obstacle characteristics and the existence of Line of Sight (LoS) communication between the sender and the receiver.

In [6], the proposed model supports the fog-based RSU architecture. The problem is modeled as an Integer Linear Programming problem. The target is to find the locations of RSUs such that their cost is reduced by deciding where to place the fog nodes as well as whether to couple or decouple the fog device to the RSU. The model supports the computation and storage capacity of the fog nodes. However, authors assume a fixed coverage radius of the RSU and no support for the signal attenuation modeling.

The RSU-opt method proposed in [7] is the closest to our work. The problem is modeled as a Mixed-Integer Linear Programming (MILP) problem. The objective is to find the optimal number and locations of RSUs with the minimum cost. Application-specific QoS constraints have to be satisfied. These constraints are expressed as the coverage percentage of the roads and the percentage of satisfied computational demand. The authors assume that the fog-based RSU is equipped with a specific computational server capacity (CPU cycles). The impact of obstacles is also modeled based on the free space path-loss and obstacle shadowing model. However, they only consider homogeneous RSUs equipped with a single processor (processor types are identical in terms of CPU cycles).

We propose a new RSU placement for VANET namely Fog-ROCL. We relax the assumptions of [7] to have heterogeneous RSUs. We utilize the concept of Satisfiability Modulo Theories (SMT) to solve a multi-objective optimization problem. Our target is to find the number, locations and processing capacity of the RSUs such that cost is minimized and the QoS is maximized. Our method generates a set of alternative solutions as Pareto front that allows the service provider for more efficient network decision.

III. Fog-based RSU Optimum Configuration and Localization

SMT solvers have proved their efficiency in solving many problems including software and hardware verification. The SMT concept is also applied in optimum IC placement [8], optimum task scheduling [9], robot motion planning [10] and many real world applications. They generate better results compared to other methodologies including those which are based on MILP. We believe that SMT is a promising approach for solving the RSU optimum distribution problem. To the best of our knowledge, we are the first to formulate and solve the RSU optimum distribution problem using SMT. In this section, we first introduce the concept of SMT that we use to formulate and solve the optimization problem. Then we describe the proposed model.

Satisfiability Modulo Theories (SMT) uses the concept of propositional Satisfiability (SAT) [11]. SAT is used to decide whether a boolean formula expressing constraints has a solution that makes it evaluate to true. If a solution is found then the problem is said to be satisfiable. Otherwise, the problem is unsatisfiable. The SAT problem input is a propositional logical formula F in CNF (Conjunctive Normal Form). CNF is the conjunction of clauses. A clause is the disjunction of literals (a literal is a boolean variable or its negation). The following formula is an example of a SAT input.

$$F = (x_1 \lor x_2) \land (\neg x_1 \lor x_3) \tag{1}$$

Some problems need more expressive logic such as the first-order logic. SMT is used to decide the Satisfiability of first-order logical formula with respect to a background theory. Examples of theories include: integers, reals, arrays and bit-vectors. An SMT solver combines a SAT solver and a theory solver. The SMT solver generates assignments for the variables which are consistent with the background theory.

We choose Microsoft Z3 [12] to solve the optimization problem. Z3 is an SMT solver developed by Microsoft Research. Z3 supports many theories such as theory of real arithmetic, integer arithmetic, bit vectors, etc. However, the solution of an SMT problem is a feasible solution, not an optimal solution. After the development of νZ [13], an extension to Z3 solver developed by Microsoft Research that handles the optimization problems, solving the optimization problems using Z3 became possible. νZ also supports multi-objective optimization in three different modes: lexicographic, independent objectives and Pareto front [13]. We use νZ in our work to generate the Pareto front that represents the set of the best trade-off between cost and the achieved QoS.

We consider a scenario of a smart city where vehicles are moving in urban area as illustrated in Fig. 1. Our assumptions are:

1. Each vehicle runs an application and sends data collected by sensors to the road-side units (RSUs) for processing.
2. Vehicles are connected directly to the RSUs (V2I communication) through DSRC.
3. A fog node is an RSU equipped with a computational server. In the rest of this paper, it is referred to as "RSU". The RSU is connected to the internet where a cloud server is available for further data processing.

We consider the target area as a set of N square cells in the x-y plane. Initially, each cell has a size of L x L where L is the side length of the square cell in meters. Each cell is a potential location of a single RSU. The location of the RSU within the cell is chosen such that it is the nearest point to the center and located on a road. The adjacency matrix $A_{i,j}$ describes whether RSU deployed at cell i covers the roads inside cell j or not. The computational demand in each cell D_i is described in terms of the maximum number of requested messages to be processed per second in cell i. We assume that each RSU is equipped with a single CPU and the RSUs are not homogeneous. The processors in RSUs are not identical in terms of CPU computation power. We describe the computational capacity of the RSU in terms of the total number of messages the RSU processor can process per second. If a new message arrives at the RSU when the processor is busy processing other messages, the message is dropped. The number of dropped messages is a QoS metric. Higher number of dropped messages denotes poor QoS.

The QoS is described as the road coverage percentage of the area α and the percentage of the satisfied computational demand γ. Our target is to find the optimal configuration of RSUs such that α and γ are maximized with minimum cost. However, there is a minimum acceptable level of QoS that has to be satisfied, α_{th} and γ_{th}. Table I summarizes the notation used in the problem formulation.

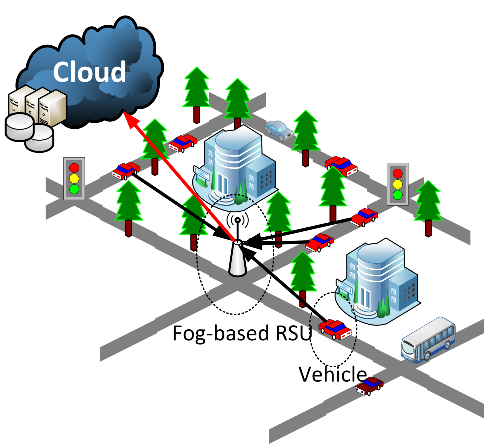

Fig. 1. Smart city scenario.

The problem is formulated as a multi-objective optimization problem and it is solved based on the concept of Satisfiability Modulo Theories (SMT) where boolean satisfiability is combined with theory of integer arithmetic. Microsoft Z3 is used as the solver with the help of νZ in Pareto front mode. The solution represents the best trade-off among the objectives. The problem is formulated as follows:

$$\min \quad \sum_{i=0}^{N-1} CY_i + \sum_{i=0}^{N-1} \sum_{k=0}^{P-1} a_k X_{i,k} \tag{2}$$

$$\max \quad \alpha = \left(\frac{\sum_{j=0}^{N-1} H_j L_j}{\sum_{j=0}^{N-1} L_j}\right)\% \tag{3}$$

$$\max \quad \gamma = \left(\frac{\sum_{j=0}^{N-1} H_j D_j}{\sum_{j=0}^{N-1} D_j}\right)\% \tag{4}$$

$$\alpha \geq \alpha_{th} \tag{5}$$

$$\gamma \geq \gamma_{th} \tag{6}$$

$$H_j = \bigvee_{i=0}^{N-1} Z_{j,i} \qquad \forall j \tag{7}$$

$$Y_i = \bigvee_{k=0}^{P-1} X_{i,k} \qquad \forall i \tag{8}$$

TABLE I
FOG-ROCL NOTATION FOR PROBLEM FORMULATION

Symbol	Type	Description
N	int	Number of grid cells
P	int	Number of CPU types
C	int	RSU fixed cost
a_k	int	Cost of CPU type k
m_k	int	Max. processing capacity of CPU type k
$X_{i,k}$	bool	indicates if a CPU of type k is deployed inside the RSU deployed at cell i
Y_i	bool	Indicates if there is a RSU deployed at cell i
$Z_{j,i}$	bool	Indicates that a RSU is deployed at cell i and this RSU covers cell j in terms of both coverage and computational demand
H_j	bool	Indicates if cell j is covered
A	matrix of bool	The adjacency matrix. $A_{i,j}$ indicates which cells j are covered if a RSU is deployed at cell i
D_j	int	Demand at cell j
L_j	int	Total length of roads at cell j
α	int: [0, 100]	Coverage Percentage
γ	int: [0, 100]	Percentage of satisfied demand
α_{th}	int: [0, 100]	Minimum acceptable coverage percentage
γ_{th}	int: [0, 100]	Minimum acceptable percentage of the satisfied demand

$$Z_{j,i} \rightarrow Y_i \wedge A_{i,j} \qquad \forall i,j \tag{9}$$

$$\sum_{j=0}^{N-1} D_j Z_{j,i} \leq \sum_{k=0}^{P-1} m_k X_{i,k} \qquad \forall i \tag{10}$$

$$\sum_{k=0}^{P-1} X_{i,k} \leq 1 \qquad \forall i \tag{11}$$

The objectives of the optimization problem is represented by (2), (3) and (4). Equation (2) describes the total cost. The first term represents the total fixed cost while the second term represents the total cost of the CPUs inside all of the RSUs. Equation (3) describes the road coverage percentage. Equation (4) describes the percentage of the satisfied demand. The constraints are represented by (5)-(11). Equations (5) and (6) ensure that a minimum level of QoS is achieved. Equation (7) sets H_j to true if there is any RSU at any cell i that covers cell j. Equation (8) describes that cell i has an RSU if there is any CPU of any type k deployed at cell i. For $Z_{j,i}$ to be true, there are two constraints that must be fulfilled. The first constraint is described by the implication relation in (9). This relation ensures that if $Z_{j,i}$ is true then there must be a RSU at cell i (Y_i is true) and the signal transmitted by this RSU covers cell j ($A_{i,j}$ is true). The second constraint is described by (10) which ensures that the total computational demand generated by vehicles at cell j does not exceed the total processing capacity of the RSU deployed at cell i. Equation (11) ensures that the RSU is equipped with a single CPU of type k at maximum.

IV. Experimental Setup

Our methodology is divided into 3 main modules: the pre-processing module, the optimization module and the evaluation module.

A. The pre-processing Module

A map of Cairo city is used as the target area to evaluate the proposed solution. Data about roads and obstacles are obtained from OpenStreetMap[1]. The map covers 2.2 x 1.9 square kilometers. We used SUMO (Simulation of Urban MObility) framework [14] and its tools to extract the network as a set of line segments and the obstacles as polygons. We wrote a script using Python with the help of Shapely[2], a python package for manipulation and analysis of geometric objects, to divide the map into a set of cells in the x-y Cartesian coordinates. Each cell has a candidate location for an RSU. The exact candidate RSU location within a cell is chosen such that it is the nearest point to the center that is located on a road.

Due to the obstacles in each cell (buildings and walls attenuating the signals between vehicles and RSUs), there are some cells that cannot be covered by only one RSU. In such case, each cell is divided into 4 smaller square cells of equal sizes. The division process is done recursively until each cell is covered by a single RSU. Fig. 2 describes how the cell is divided. Shaded cells are excluded from the set due to lack of roads. Dots represent candidate locations for the RSUs.

To calculate the adjacency matrix, the free space path loss propagation model combined with obstacle shadowing proposed in [15] is applied to calculate the coverage range of the RSUs. We set $A_{i,j}$ to true if at least 90% of the vehicles in cell j can receive a signal above the receiver's sensitivity transmitted from cell i.

B. The Optimization Module

To solve the optimization problem, we have to calculate the computational demand in each cell in terms of maximum number of generated messages per second. We used SUMO floating car data (FCD) output to calculate the maximum number of vehicles in each cell from the traffic trace. A random traffic trace is generated using the SUMO randomTrips tool where the scenario runs for 200s. The number of vehicles in each cell is mapped into computational demand by multiplying the number of vehicles by the number of messages generated by a single vehicle per second.

C. The Evaluation Module

To evaluate our proposed Fog-ROCL method, we use the simulation framework Veins [16] version 4.7.0. Veins is an open source framework that integrates OMNet++ network simulator with SUMO traffic simulator.

D. Methodology

We solve the optimization problem using Microsoft Z3 using its Python APIs. All instances run for 200 seconds using the same traffic trace. We set the values of both α_{th} and γ_{th} to 70% and the cell size L to 400m. We use simulation parameters similar to [7]. Transmission power is set to 21 dBm, receiver's sensitivity is set to -100 dBm, message size is set to 160 bytes and message exchange frequency is set to 1 Hz. It is assumed that:

1. There are 4 different types of processor capacities: 25, 50, 100 and 150 messages/s (type 0, type 1, type 2, type 3 respectively).

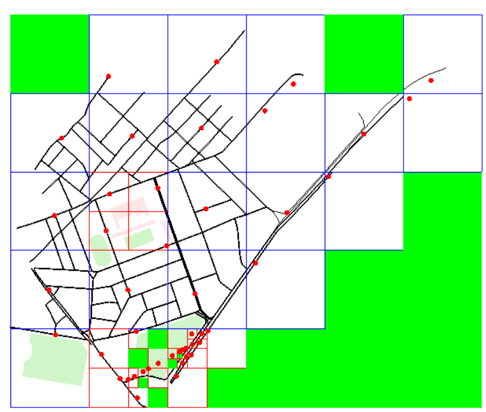

Fig. 2. The map division process.

[1] https://www.openstreetmap.org

[2] https://pypi.org/project/Shapely

2. The RSU cost = the RSU fixed cost + total cost of processors equipped inside the RSU.
3. The same fixed cost for all RSUs of 500$ each.
4. The processor costs are 100, 200, 400 and 600$.

The solver generates Pareto front that represents the best trade-off among: number of RSUs (R), cost, coverage percentage (α) and percentage of satisfied demand (γ). The solver also generates the locations of RSUs.

For comparison, we consider the RSU-opt method proposed in [7] and described in section II. The RSU-opt method generates a single solution per run. We set coverage percentage and computational demand both to 100% and solve the problem based on their proposed model using CPLEX[3]. We compare the solution obtained by CPLEX with the Pareto solution generated by Fog-ROCL that gives 100% coverage and 100% satisfied demand. Since the RSU-opt method supports a single CPU type, we run the simulation for 2 runs with different CPU capacities. In the first run it is set to 25 message/s while in the second run it is set to 100 message/s. The following are the metrics used for the evaluation:

- The packet-loss (percentage): the number of messages that are not successfully received by the receiver. A better solution has a lower packet-loss value.

$$\text{Packet loss} = \frac{\text{messages sent - messages received}}{\text{messages sent}}\% \tag{12}$$

- Dropped-messages (percentage): the number of messages that are received by the RSU but could not be processed due to lack of available CPU capacity. A better solution has a lower drop value.

$$\text{Dropped messages} = \frac{\text{messages dropped by RSUs}}{\text{messages generated by vehicles}}\% \tag{13}$$

- The total RSU cost ($): a better solution has lower cost.
- Number of RSUs: a better solution has a fewer number of RSUs; as the number of RSUs increases, the interference between signals increases. Hence, deployment of a fewer number of RSUs that gives the same QoS is better.

V. Results

We first study the impact of cell size on the execution time. Table II compares the impact of three different cell sizes: 400m, 600m and 800m. Dividing the map into smaller cells is more accurate. However, it consumes more time during preprocessing to calculate the adjacency matrix. Smaller cells result in larger number of candidate locations which increases the size of the adjacency matrix. Larger cells result in less number of locations. However, they are subjected to multiple and recursive division process especially in obstacle dense cells. On the other hand, Z3 consumes much time as the number of candidate location increases because the solver has to search for the optimal solution among a larger set of possible locations.

Table III shows a comparison among the results obtained by the RSU-opt method proposed in [7] and our proposed

[3]https://www.ibm.com/products/ilog-cplex-optimization-studio

TABLE II
Impact of cell size on the execution time of Z3.

Cell size (m^2)	400x400	600x600	800x800
# Initial cells (Before division)	22	10	8
# Candidate locations (After division)	45	39	22
Execution time (s) (Preprocessing)	52	47	23
Execution time (s) (Z3)	20	13	0.7

TABLE III
Simulation results for RSU-Opt and Fog-ROCL for 100% solution.

	RSU-opt (25 msg/s)	RSU-opt (100 msg/s)	Fog-ROCL
# RSUs	12	9	9 (7 of type 0, 1 of type 2 and 1 of type 3)
Cost	7200 $	8100 $	6200 $
Packet-loss	1%	0.7%	0.9%
Dropped messages%	35%	0.0%	0.001%

Fog-ROCL method. Cell size is set to 400m and the number of initial candidate locations is 45. The 100% solution (100% coverage and 100% satisfied demand) is obtained using RSU-opt by 12 RSUs deployed with total cost of 7200 $ and CPU type of capacity 25 messages/s. While it is obtained using the same RSU-opt method by 9 RSUs deployed with total cost of 8100 $ and CPU type of capacity 100 messages/s. There is a great enhancement in the dropped messages% (from 35% to almost 0%) as the RSU capacities increased and can handle much traffic demand than the RSUs with 25 messages/s capacity. However, increasing the capacities of all RSUs is not cost effective as less capacities may be sufficient for some cells with no need to increase their RSU computational capacities.

The last column in Table III illustrates the results of our proposed Fog-ROCL method. Both α_{th} and γ_{th} are set to 70%. The 100% solution is obtained using Fog-ROCL by 9 RSUs deployed with total cost of 6200 $. Results obtained by Fog-ROCL outperform results obtained by RSU-opt with CPU capacity set to 25 messages/s in terms of dropped message percentage (almost 0.0% vs. 35%). This enhancement is obtained as the computational capabilities of each RSU is assigned based on the computational demand generated in its coverage area. Assignment of identical CPUs of 25 messages/s computational capacity in RSU-opt raises the dropped percentage to 35% as the computational demand in some cells exceeds the RSU capability. Fog-ROCL is more cost effective than RSU-opt with capacity set to 100 messages/s (8100 $ vs. 6200 $) while the packet loss and dropped messages values are very much close. As a result, the proposed Fog-ROCL method is better than the RSU-opt method. The RSUs capacities are selected from a pool of different capacities such that they satisfy the different computational demand in each cell with less cost than the RSU-opt method.

Fog-ROCL generates 12 different solutions in the Pareto set, not only the 100% solution, in 20 seconds. Due to space limit, we cannot list all of the 12 solutions. However, some of the Pareto front solutions are listed in Table IV. For instance, solution number 1 achieves 100% for both coverage and satisfied demand by the deployment of 9 RSUs of cost 6200$. Seven RSUs out of the 9 are equipped with a single CPU of type 0, One RSU is equipped with a single CPU of type 2, and one RSU is equipped with a CPU of type 3. We test solution #3 in Table IV (86% for coverage and 82% for demand) with cost 1600$ vs. the 100% solution with 6200$. Results are illustrated in table V. The packet loss increased from 0.9% to 2% and the dropped messages increased from 0.001% to 1.7%. The service providers have to select the best trade-off according to their criteria. As the size of Pareto front increases in larger instances, it would not be feasible for the user to manually select a single solution among the generated set. As a result, we recommend the automation of the post-Pareto analysis based on the user-defined criteria, which is part of our future work.

VI. Conclusion

This paper discusses the problem of optimal deployment and localization of fog-based RSUs in urban area. We propose Fog-ROCL, a model that solves the problem as a multi-objective optimization problem using the concept of SMT to solve the problem. The objective is to obtain a cost-effective configuration that maximizes the QoS represented in coverage percentage and the satisfied computational demand. We apply our proposed method to the Cairo city scenario and it is evaluated against the RSU-opt technique that uses CPLEX as the optimizer. Our method shows better results in terms of RSU deployment cost and percentage of dropped messages due to lack of computational capabilities. Our method generates multiple alternatives as Pareto front for the end user. Our future work includes the post-Pareto analysis of the obtained results as well as studying the impact of the cell size on the output and how to select its value.

TABLE IV
PARETO FRONT OBTAINED BY FOG-ROCL.

#	#**RSUs**	**Cost** ($)	**Coverage** (%)	**Satisfied demand** (%)
1	9 [7 X type 0, 1 X type 2, 1 X type 3]	6200	100	100
2	4 [2 X type 0, 1 X type 2, 1 X type 3]	3200	98	99
3	2 [1 X type 1, 1 X type 2]	1600	86	82

TABLE V
COMPARISON BETWEEN SOLUTIONS #1 AND #3 GENERATED BY FOG-ROCL.

Solution #	#1	#3
# RSUs	9 (7 of type 0, 1 of type 2 and 1 of type 3)	2 (1 of type 1 and 1 of type 2)
Cost	6200 $	1600 $
Packet-loss	0.9%	2%
Dropped messages%	0.001%	1.7%

References

[1] R. A. Uzcategui, A. J. De Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009.

[2] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz, "Wireless communication technologies for ITS applications [topics in automotive networking]," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 156–162, May 2010.

[3] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *2014 Federated Conference on Computer Science and Information Systems*, Sep. 2014, pp. 1–8.

[4] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei, H. He, and Q. Yang, "Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140–2150, Oct 2017.

[5] S. Mehar, S. M. Senouci, A. Kies, and M. M. Zoulikha, "An optimized roadside units (RSU) placement for delay-sensitive applications in vehicular networks," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 121–127.

[6] C. Yu, B. Lin, P. Guo, W. Zhang, S. Li, and R. He, "Deployment and dimensioning of fog computing-based internet of vehicle infrastructure for autonomous driving," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 149–160, Feb 2019.

[7] G. Premsankar, B. Ghaddar, M. Di Francesco, and R. Verago, "Efficient placement of edge computing devices for vehicular applications in smart cities," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–9.

[8] S. M. Saif, M. Dessouky, M. W. El-Kharashi, H. M. Abbas, and S. M. Nassar, "A platform for placement of analog integrated circuits using satisfiability modulo theories," *Journal of Circuits, Systems, and Computers*, vol. 25, 2016.

[9] A. Malik, C. Walker, M. O'Sullivan, and O. Sinnen, "Satisfiability modulo theory (SMT) formulation for optimal scheduling of task graphs with communication delay," *Computers and Operations Research*, vol. 89, pp. 113 – 126, 2018.

[10] F. Imeson and S. L. Smith, "An SMT-based approach to motion planning for multiple robots with complex constraints," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 669–684, June 2019.

[11] L. de Moura and N. Bjørner, "Satisfiability modulo theories: An appetizer," in *Formal Methods: Foundations and Applications*, M. V. M. Oliveira and J. Woodcock, Eds. Springer Berlin Heidelberg, 2009, pp. 23–36.

[12] L. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Berlin Heidelberg, 2008, pp. 337–340.

[13] N. Bjørner, A.-D. Phan, and L. Fleckenstein, "νz - an optimizing smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Berlin Heidelberg, 2015, pp. 194–199.

[14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 2575–2582.

[15] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments," in *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, Jan 2011, pp. 84–90.

[16] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 1 2011.