



HAL
open science

Green Edge Servers Placement for Intelligent Transport Systems

Sabri Khamari, Abdennour Rachedi, Toufik Ahmed, Mohamed Mosbah

► **To cite this version:**

Sabri Khamari, Abdennour Rachedi, Toufik Ahmed, Mohamed Mosbah. Green Edge Servers Placement for Intelligent Transport Systems. 2022 13th International Conference on Network of the Future (NoF), Oct 2022, Ghent, France. pp.1-8, 10.1109/NoF55974.2022.9942580 . hal-04476077

HAL Id: hal-04476077

<https://hal.science/hal-04476077>

Submitted on 24 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Green Edge Servers Placement for Intelligent Transport Systems

Sabri Khamari
Univ. Bordeaux, Bordeaux INP,
CNRS, LaBRI, UMR5800,
F-33400 -Talence, France
sabri.khamari@u-bordeaux.fr

Abdenmour Rachedi
Univ. Bordeaux, Bordeaux INP,
CNRS, LaBRI, UMR5800,
F-33400 -Talence, France
abrachedi@u-bordeaux.fr

Toufik Ahmed
Univ. Bordeaux, Bordeaux INP,
CNRS, LaBRI, UMR5800,
F-33400 -Talence, France
tad@labri.fr

Mohamed Mosbah
Univ. Bordeaux, Bordeaux INP,
CNRS, LaBRI, UMR5800,
F-33400 -Talence, France
mohamed.mosbah@u-bordeaux.fr

Abstract— Edge computing empowers service providers to deploy smart vehicles applications that require high throughput and extremely low latency. In this context, optimal Edge servers' placement becomes more difficult since it requires addressing several interrelated requirements at the same time, such as delay, deployment cost, and energy consumption. This paper studies optimal Edge server placement for energy efficiency. The proposed approach, called Green Optimal Edge Server Placement (GOESP), models the placement problem using integer linear programming to address the trade-off between latency, energy, and deployment cost while considering Edge servers' capacity and expected vehicle's traffic on the road. GOESP minimizes the energy consumption by minimizing the number of deployed Edge servers while meeting end-to-end communication latency and avoiding servers' overloading. We evaluate the efficiency of our approach mathematically and through simulations utilizing real-world traffic extracted from open data of Bordeaux city, France. The results demonstrate that our technique outperforms other methods in terms of energy efficiency and guarantees latency and workload balancing requirements.

Keywords— Green computing, Edge computing, Edge server placement, Intelligent Transport Systems.

I. INTRODUCTION

Smart vehicle networks have become essential to deploy innovative transportations services such as dangerous situation warning, special vehicle prioritization, and area reservation. Also known as Cooperative Intelligent Transport Systems (C-ITS), smart vehicle networks improve traffic management and road safety policies, ensure a future with fewer road accidents, and enable optimal driving to reduce fuel consumption.

New C-ITS use cases, such as autonomous and remote driving, require intensive AI-based processing and video analytics and necessitate guaranteed Quality of Service (QoS) performance, including ultra-low latency, near-perfect reliability, and extremely high bandwidth [1]. Such use cases generate massive amounts of data and require in-network processing capabilities. For example, smart vehicles are equipped with Lidar, radar, and cameras to collect information in the vehicle's vicinity [2].

However, on-board vehicles' computational resources are insufficient to handle service processing tasks under a high volume of generated data. As a result, some tasks should be offloaded to remote computational nodes [3]. Cloud computing enables the provision of on-demand extensive processing and

storage resources capable of managing such enormous amounts of data. Nevertheless, cloud computing capabilities are often offered by data centers located far from the vehicle's position. As a result, end-to-end data transmission often experiences considerable delays [4], and networks frequently face bottlenecks. Additionally, transmitting high volumes of data requires high bandwidth usage that consumes a substantial amount of energy [4].

MEC (Multi-Access Edge Computing) promises to deliver adequate processing capacity while meeting delay constraints and optimizing energy consumption [4]. It entails installing a set of high-performance computing nodes near vehicles on the base stations of cellular operators or the Roadside Units (RSU) of road operators. MEC offers the efficient fulfillment of stringent QoS requirements for a diverse set of services and use cases and the handling of computationally intensive tasks. However, deploying edge servers in urban vehicular networks remains a significant challenge. Indeed, optimal edge server placement implies limiting the number of edge nodes required to meet the service level quality. Such deployment should ensure that appropriate computation and storage resources are always available to meet the C-ITS services' latency, bandwidth, and reliability requirements.

Additionally, optimal placement options are necessary to reduce deployment costs and increase energy efficiency. Inefficient Edge server placement leads to high latency and significantly imbalanced workload distribution across Edge servers. Thus, some Edge servers may be overburdened while others are underutilized. Deploying more edge nodes results in increased deployment costs and reduced energy efficiency.

This study aims to address the issue of efficiently deploying an optimal number of Edge servers while adhering to deployment costs and network performance constraints. This problem is exacerbated in metropolitan environments, where a high volume of data is generated due to user mobility and density.

This paper introduces the Green Optimal Edge Server Placement (GOESP) approach, a mathematical optimization model for a green placement of edge servers in strategic locations. Toward this objective, we formulate the problem as integer linear programming that minimizes the monetary and energy cost associated with the deployment while simultaneously satisfying a specific latency threshold and balancing workload among Edge servers. To provide a realistic

Edge server placement solution, we propose a methodology based on open data from real-world automobile traffic in Bordeaux city [6]. We perform extensive simulations to analyze the suggested approach and demonstrate its effectiveness.

The rest of this paper is organized as follows. Section II introduces some background on Multi-access Edge computing and its application in vehicular networks. Section III presents related work in edge server deployment and energy efficiency solutions. In Section IV, we present our solution and model for the green edge servers' placement. Next, Section V presents the simulation setup and performance evaluation. Finally, we briefly conclude the work and introduce future work in Section VI.

II. BACKGROUND

A. Vehicular applications

In the context of IoT Automotive, V2X (Vehicle to Everything) requires a critical communication infrastructure where reliability and ultra-low latency are determinant factors [7]. V2X covers Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), vehicle to Pedestrian (V2P). Intelligent Transportation System (ITS) applications can be categorized into road traffic safety, road traffic efficiency, and value-added services [8][9]. Use cases in these categories include collision warning, assistance (Intersection Movement, Left Turn, cooperative merging), congestion warning, etc. [8]. Communication and computation capabilities must be improved to operate an efficient and reliable vehicular application that could include real-time traffic monitoring, continuous sensing, and cooperative and augmented vision [10]. However, the current network and infrastructure based on cloud computing cannot fully satisfy the application's requirements. Recently, MEC (Multi-access Edge Computing) has emerged as an innovative architecture that offers a higher level of flexibility and provides a viable solution that can accelerate the development of ITS applications [11].

B. Multi-access Edge Computing (MEC)

Mobile Edge Computing was introduced by the European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) as a means of extending intelligence to the edge of the network along with higher processing and storage capabilities [12]. From 2017, the ETSI industry group renamed it to Multi-Access Edge Computing (MEC) since the benefits of MEC technology reached beyond mobile towards Wi-Fi and fixed access technologies. The principle of MEC consists of extending cloud computing capabilities to the edge of networks to minimize network congestion and improve resource optimization, user experience, and overall network performance.

C. Deployment of MEC systems

The primary motivation of Multi-access Edge Computing (MEC) is to bring the Cloud computing capability close to the network edges to mitigate cloud computing limits such as high latency and throughput bottleneck. Selecting MEC locations, especially Edge servers, remains the first step towards building up the MEC system and deploying vehicular applications.

However, the site selection for the deployment of Edge servers presents an important challenge. The server's placement strategy differs from the conventional base station site selection's problem. The optimal placement of Edge servers is related to the computational resource provisioning and constrained by the deployment budget [13] and vehicles traffic on roads. The MEC providers should consider computational demands of task arrival and deployment costs such as site rentals to make the MEC servers placement strategy cost-effective. In general, considering the system deployment budget, more MEC servers should be installed in regions with higher computation demands. Using existing telecom infrastructure, such as base stations, could reduce costs, but it would not solve all the problems. Indeed, due to the continuous increase in computation requirements for vehicles and various smart devices (IoT), application requirements cannot be guaranteed. Therefore, Edge servers must be deployed with a more distributed strategy and closer to the user.

III. RELATED WORK

Existing research works on the usage of edge servers, such as [14][15], focus on the placement challenge and load balancing, with little emphasis on meeting QoS requirements and on green issues such as energy consumption and deployment costs. The authors of [16] tackled the placement problem using a mixed-integer linear programming formulation that minimizes the deployment cost of Edge computing devices by jointly satisfying a target level of network coverage and computational demand. However, this work disregarded the latency requirements of vehicular applications. The work of [17] investigated the problem of deploying heterogeneous edge servers for response time minimization in mobile edge-cloud computing systems. Two stages are proposed: an offline stage using an Integer Linear Programming (ILP) technique to produce an optimal solution of edge server placement, and an online stage that deals with the mobility of users using a game theory-based scheme of base stations. Wang et al. [18] investigated the Edge server placement problem in mobile Edge computing environments for smart cities. They formulated the problem as a multi-objective constraint optimization problem that places Edge servers in strategic locations to balance workload between Edge servers and minimize Edge server access delay. Even though this work did not particularly address the vehicular network, it is the most pertinent to the placement problem discussed in this paper. We consider it for comparison with our proposed solution.

Green edge solutions for vehicular environments focus on tasks and computational resources offloading [20][21]. The authors of [3] proposed a collaborative edge computing scheme that enables vehicle-as-an-edge technology for task processing in the vehicular Internet of Things (IoT). Ning et al. [22] presented an incentive-based optimal computation offloading scheme that maximizes the service provider's utilities while reducing smart device energy consumption and task execution time. However, none of these works considers the placement of edge servers as a vital parameter for green edge solutions for automotive networks.

Our study focuses on developing a global solution that considers both the placement strategy required to meet QoS requirements and the energy efficiency limitations associated with reducing the deployment cost.

IV. GREEN OPTIMAL EDGE SERVER PLACEMENT

Figure 1 depicts the proposed reference architecture for vehicular green edge computing. We consider connected vehicles equipped with ITS-G5, cellular communication interfaces, or both. These vehicles collect data from their surrounding environment. Frequently, advanced processing is often required by vehicle services' tasks. To meet the requirements of C-ITS applications, storage and computing capabilities are provided at the network edge close to the vehicles. The edge layer comprises edge servers that can run on renewable energy sources such as solar energy if the deployment area is equipped with appropriate materials. In such a scenario, vehicles offload their processing jobs to the edge servers via cellular connection (4G/5G) using the base station (BS) or via the ITS-G5 communication using roadside equipment (RSU). The edge servers are connected to the cloud and the road operator platform via the core network.

The following subsection describes our solution, namely Green Optimal Edge Server Placement (GOESP), which enables effective green edge server placement while meeting the requirements of automotive applications (latency and load balancing) and taking into account the green environmental elements of communication

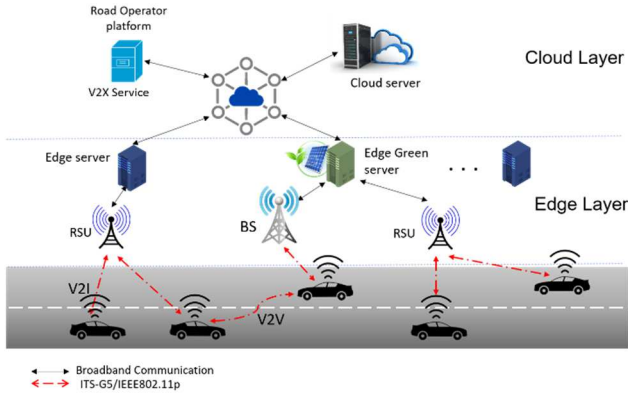


Figure 1. Vehicular green Edge computing reference architecture

A. System model

The system is composed of a set $J = \{S1, S2, \dots, Sm\}$ of potential locations for Edge servers and a set $I = \{R1, R2, \dots, Rn\}$ of roadside units and/or BS. Some of the proposed sites include renewable green energy sources like solar power. Figure 2 illustrates an example of the vehicular edge computing system where one of the three edge servers ($S2$) is deployed at a location with a green energy source.

Assume that K Edge servers will be assigned to K different locations, where K is a constant (in Figure 2, $K=3$). To process the requests sent by the vehicles, we suppose that each Edge server has the same restricted computing resource (storage, processing, and memory). Each RSU/BS is linked to one Edge server to delegate vehicle requests.

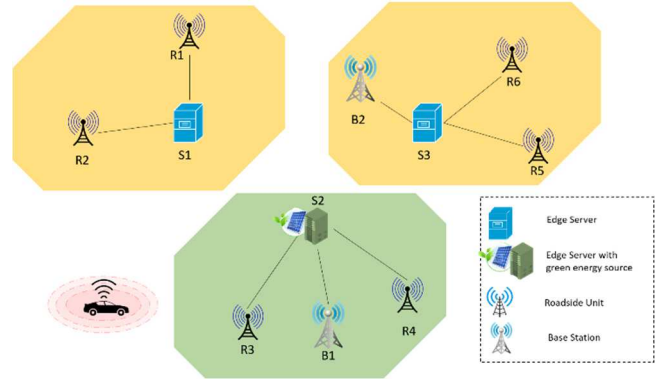


Figure 2. Example of the vehicular green edge computing system

The objective is to install as few edge servers as possible to minimize energy consumption while serving all vehicle requests without surpassing the servers' capacity. Additionally, the model emphasizes the importance of choosing a location in an area with a green energy supply.

The notations utilized in our model are listed in Table 1. c_j presents the cost of deploying an Edge server at location j . This cost combines the monetary and energy consumption cost; hence, a lower cost is attributed to a location with a green energy source. The tuple memory, storage, and processing express the Edge server capacity. Similarly, each RSU/BS has its requirements or demands described similarly. Such demand represents the aggregated tasks of vehicles received by this RSU/BS.

TABLE 1. NOTATIONS

Symbol	Description
$V = I \cup J$	I : a set of RSU/BS J : a set of the potential location of Edge servers.
$G = (V, E)$	E : a set of links between roadside units and Edge servers
n	Number of roadside units
m	Number of potential locations for Edge servers
K	The number of deployed Edge servers
c_j	Cost of deployment of Edge server at location j
L_{max}	Maximum latency (proportional to the distance)
$L_{i,j}$	The latency between RSU i /BS i and Edge server location j
Pr_i	Processing demands of RSU i /BS i
Mr_i	Memory demands of RSU i /BS i
Sr_i	Storage demands of RSU i /BS i
Ps_j	Processing capacity of server j
Ms_j	Memory capacity of server j
Ss_j	Storage capacity of server j
x_j	Binary variable for Edge server placed at location j
$y_{i,j}$	Binary variable for RSU i /BS i linked to Edge server at location j
Z	The total cost of Edge server's deployment

Accordingly, we formulate an integer linear programming model that captures the model's features under consideration. We begin by defining the decision variables of the model:

$$x_j = \begin{cases} 1, & \text{if server is deployed at location } j \\ 0, & \text{else} \end{cases} \quad (1)$$

$$y_{i,j} = \begin{cases} 1, & \text{if RSU } i \text{ is linked to server } j \\ 0, & \text{else} \end{cases} \quad (2)$$

$$Z = \sum_{j \in J} c_j x_j : \text{ total cost} \quad (3)$$

The objective function is defined as:

$$\text{Min } Z, \text{ i.e. } \text{Min } \sum_{j \in J} c_j x_j \quad (4)$$

s.t

$$\sum_{j \in J} y_{i,j} x_j = 1, \quad \forall i \in I \quad (5)$$

$$\sum_{j \in J} L_{i,j} y_{i,j} x_j \leq L_{max}, \quad \forall i \in I \quad (6)$$

$$\sum_{i \in I} Pr_i y_{i,j} \leq Ps_j x_j, \quad \forall j \in J \quad (7)$$

$$\sum_{i \in I} Mr_i y_{i,j} \leq Ms_j x_j \quad \forall j \in J \quad (8)$$

$$\sum_{i \in I} Sr_i y_{i,j} \leq Ss_j x_j \quad \forall j \in J \quad (9)$$

$$x_j \in \{0,1\} \quad \forall j \in J \quad (10)$$

$$y_{i,j} \in \{0,1\} \quad \forall j \in J, \forall i \in I \quad (11)$$

Beginning with the decision variables in (1) and (2), we describe each expression in our model. x_j denotes a Boolean variable indicating whether a candidate position is selected for Edge server deployment. $y_{i,j}$ represents a Boolean variable that shows which Edge server is connected to each RSU/BS. For instance, $y_{5,2} = 1$ indicates that RSU "5" is connected to the Edge server deployed in position "2". The objective function in (4) aims to minimize the deployment cost by reducing the number of Edge servers to be deployed where c_j denotes the deployment cost of Edge server at location j from J .

Following that, we discuss the constraints. Equation (5) ensures that each roadside unit is connected to only one server among the deployed servers. Constraint (6) guarantees that each RSU is connected to an Edge server without exceeding the maximum tolerated latency and L_{max} is defined by distance. Constraints (7-9) ensure that memory, storage, and processing requirements are met. It guarantees that the total number of requests sent by roadside units and base stations to the Edge server does not exceed its capacity. Finally, constraints (10-11) ensure that the decision variables satisfy the integrality requirements.

B. Use case Scenario

We consider the city of Bordeaux, France, as the target area for Edge servers' placement. A map of Bordeaux city, including information about roads and buildings, is obtained

from OpenStreetMap [23]. The map bounds an area of 15.10 km². Figure 3 shows the concerned area. For simplicity but without loss of generality, we assume that 27 roadside units are installed at road intersections at a set spacing of around 800 meters apart. We adopted this strategy because of the lack of data on actual RSU deployment. The following stage consists of identifying potential Edge server sites. We chose those locations using a straightforward technique. Between every three or four RSUs, one site is specified as a possible location for an Edge server. The optimization model chooses the most strategic places for deploying Edge servers based on a set of 19 possible Edge server locations. The candidate locations shown in a green square in Figure 3 are equipped with a renewable energy source.

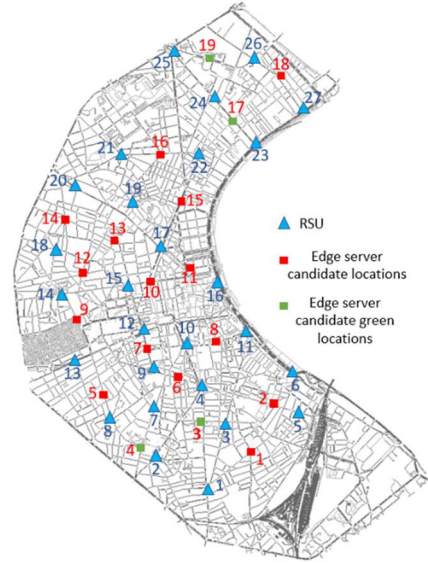


Figure 3. Use case scenario: Edge servers' placement on Bordeaux

We used the open dataset OpenDataBordeaux [6] to generate a traffic trace similar to the real one of Bordeaux. Based on this data, we identified three classes of roads: high traffic roads, moderate traffic roads, and low traffic roads. The RSUs send vehicles' tasks to the edge servers according to generated traffic trace. Therefore, we distinguish three classes of RSU: RSU dealing with high demands, RSU with moderate demands, and RSU with low demands, as shown in Table 2.

TABLE 2. RSU CLASSIFICATION

Category	RSU
High demand	R9, R10, R12, R23
Moderate demand	R3, R4, R5, R6, R7, R11, R15, R16, R17, R22
Low demand	R1, R2, R8, R13, R14, R18, R19, R20, R21, R24, R25, R26, R27

V. EVALUATION AND RESULTS

A. Methodology and simulation setup

Figure 4 provides a comprehensive overview of the methodology and simulation setup used to evaluate our green

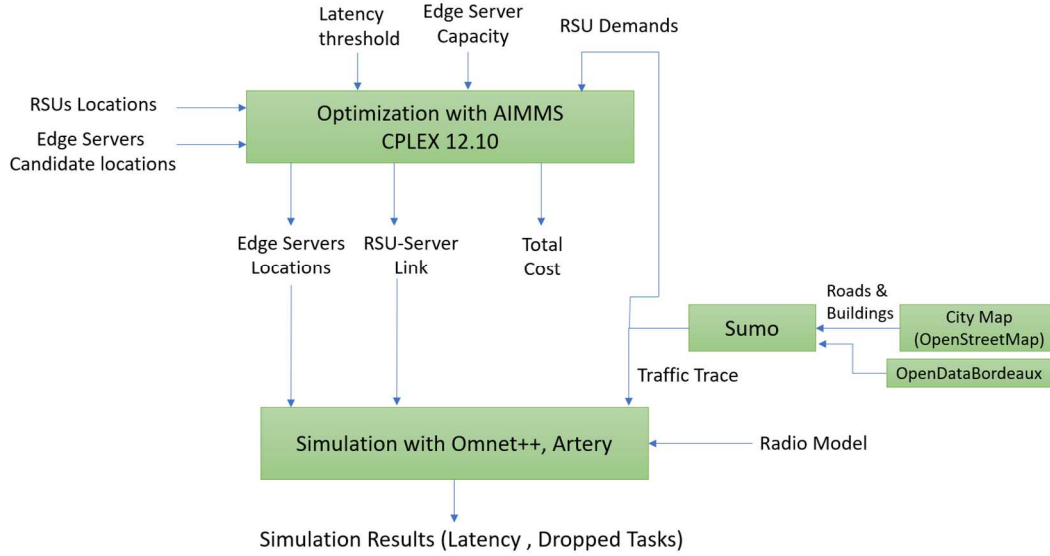


Figure 4: Overview of the used methodology to evaluate “GOESP”.

placement strategy. To obtain the optimal solution of our model, we use AIMMS (Advanced Interactive Multidimensional Modeling System [24]) with CPLEX12.10 solver. AIMMS is a perspective analytics platform that allows the modeling and development of optimization-based applications. The model requires the following inputs: the geographical coordinates of roadside units, the geographical coordinates of the Edge server's candidate locations, the latency threshold (represented by distance), Edge servers' capacity, and RSUs demands (in terms of processing, memory, and storage). According to the model, the latency, and servers' capacity constraints, each roadside unit is linked to one of the edge servers. As a result, the solver returns the number of Edge servers to deploy as well as the placements of the servers among the possible locations. Therefore, the total cost of deployment is calculated. The machine for this optimization task presents the following configuration: processor intel i7-10610U CPU @ 1.80GHz 2.30 GHz and RAM 16 GO.

Once the optimal placement of edge servers is obtained, we run Omnet++ [25] and the Artery framework [26] to simulate vehicle traffic. We use SUMO [27] to generate the traffic trace of the simulation. The simulations were conducted on a computer with 16 GO RAM and the Intel i7-9700 CPU @ 3.00GHz x8 processor.

Table 3 summarizes the parameters used to find the optimal solution using AIMMS and run simulations. We use two values to represent the cost edge server deployment. It is equal to one (unit) if the candidate location has a green energy source and equal to two (unit) otherwise. We suppose that all Edge servers have the same capacity, and the demands of the RSUs depend on the category to which the RSU belongs.

TABLE 3. MODEL AND SIMULATION PARAMETERS

AIMMS parameters		
Edge server deployment cost at location i	Location with a green energy source	1 unit
	Location without a green energy source	2 unit
RSU demands (processing/Memory/Storage)	High Demand	30 unit
	Moderate Demand	20 unit
	Low Demand	10 unit
Edge server capacity (Processing/Memory/Storage)	150 unit	
Latency threshold (in distance)	1500 m	
Omnnet/Artery parameters		
RSU transmission power	47,9 mW	
RSU antenna high	5m	
Vehicle transmission power	200 mW	
Frequency band	5.9 GHz	
Propagation model	ConstantSpeedPropagation (speed of light)	
Pathloss model	GEMv2 [28]	
Obstacle loss model	DielectricObstacleLoss	
Background noise Model	IsotropicScalarBackgroundNoise (-110 dBm)	
Message size (payload)	1000 Bytes	
Message frequency	1 Hz	
Vehicles max density	729 Vehicles	

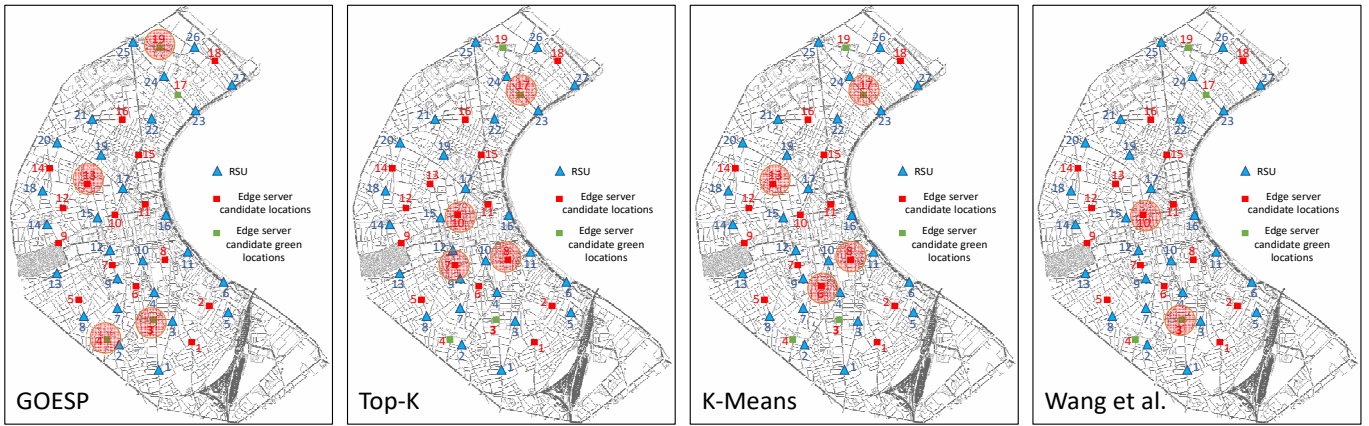


Figure 5: Edge server's placement locations for different approaches: GOESP, Top-K, K-Means and Wang et al. [18]

B. Results and performance evaluation

The optimal solution calculated with the solver CPLEX indicates that four ($K=4$) Edge servers should be deployed in the concerned area of Bordeaux. The chosen locations are location 3, location 4, location 13, and location 19, as shown in Figure 5. The placement results show that our approach favors the green communication strategy as three (3) among the four (4) locations are powered by renewable green energy. Thus, the total deployment cost is 5.

To better evaluate our solution, we implemented other placement strategies such as Random, Top-K, and K-means as baseline comparisons because they are the most used strategies in such problems. Furthermore, we also implemented Wang et al. approach [18], which is the most relevant state-of-the-art placement strategy compared to our work. The different strategies are described below:

- **Random:** this method randomly assigns the K Edge servers to candidate locations. We evaluated two different approaches to this problem, based on how the roadside units are connected to the Edge servers. The first is Random-Random, in which RSUs are randomly assigned to servers. The second is Random-Nearest, in which each RSU is connected to the Edge server located closest to it.
- **Top-K:** The K RSUs with the highest traffic demands are picked. Each RSU is connected to the Edge server located closest to it. Then, the Edge servers are deployed in the places that are closest to each RSU (K locations).
- **K-means clustering:** This technique is frequently used to automatically cluster a dataset into K groups. We discover K clusters of roadside units and then put K Edge servers in their centers using the K-means clustering technique.
- **Wang et al.'s approach [18]:** considers the deployment of edge servers in smart cities to balance the workload between Edge servers and minimize the Edge server access delay. They adopted mixed integer programming to find the optimal solution.

Table 4 below summarizes the placement results for the different strategies. We fixed the number of deployed edge servers to $K=4$ for comparative purposes for the Random, Top-

K, and K-means techniques. As a result, our solution has a lower deployment cost than Top-K and K-means. The Wang et al. approach has the lowest deployment cost by deploying just two servers. However, this solution does not ensure the system's performance, and demands may exceed the servers' capacity at peak periods.

TABLE 4. EDGE SERVERS' PLACEMENT RESULTS

Placement strategy	Edge servers' locations	Deployment cost
GOESP	Loc 3, Loc 4, Loc 13, Loc 19	5
Top-K	Loc 7, Loc 8, Loc 10, Loc 17	7
K-means	Loc 6, Loc 8, Loc 13, Loc 17	7
Wang et al. [18]	Loc 3, Loc 10	3

The link between each roadside unit and the Edge server is shown in Table 5, and the latency between the entities is stated in terms of distance. For instance, Server 1, located at position 3, is used to connect R5 at a distance of 1101 meters, R6 at a distance of 1204 meters, and so on. Additionally, we show the Edge servers' workload, which is the aggregate of the demands of the roadside units connected to this server; it is 120 units for server 1. The obtained results adhere to the server's latency and capacity limits. All RSUs (R1 to R27) are linked to one of the Edge servers without exceeding the model's maximum latency (measured in distance) of 1500m. For example, regarding the capacity constraint, the sum of RSUs linked to server 1 is 120 units, which is less than server 1's capacity (150 units). Our method satisfies the server's capacity constraint while also ensuring optimal load balancing among edge servers.

TABLE 5. EDGE SERVERS' PLACEMENT AND RSU CONNECTIONS RESULTS

Edge server	Roadside ID (distance to the edge server)
Server 1 (at location 3)	R5 (1101m), R6 (1204m), R7 (582m), R8 (1058m), R11(1117m), R12(1236m)
Server 2 (at location 4)	R1 (893m), R2(191m), R3 (1030m), R4(1005m), R9(911m), R10(1303m)
Server 3 (at location 13)	R13 (1337m), R14 (852m), R15 (516m), R16 (1175m), R17 (530m), R18 (721m), R19(496m), R20 (877m)
Server 4 (at location 19)	R21 (1338m), R22 (534m), R23 (373m), R24 (242m), R25 (1056m), R26 (740m), R27 (793m)

Figure 5 summarizes and indicates the chosen locations for deployment of Edge servers for each placement strategy. The Random-Random and Random-Nearest approaches are also not presented here as their results are random. Figure 6 highlights

the load balancing results achieved using the different placement strategies by displaying the percentage of tasks handled by each server. Our approach GOESP provides an extremely high level of load balancing where servers process nearly the same number of tasks (maximum difference of 5%). The GOESP approach enhances the fairness among the Edge servers. The other strategies do not ensure good load balancing. For example, server 2 handles 36% of tasks (160 units) in Top-K, surpassing 150 units.

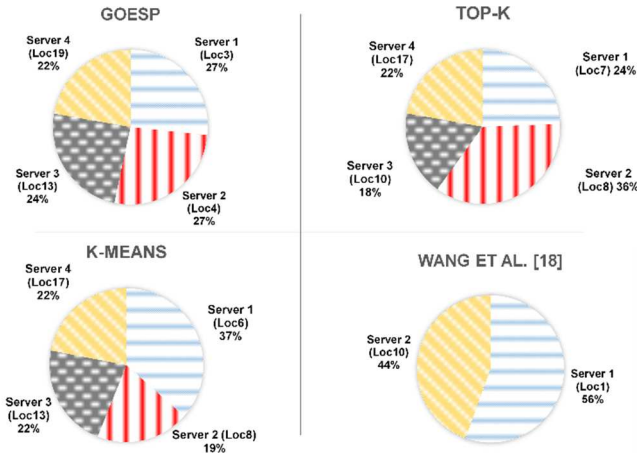


Figure 6. Workload results of GOESP, Top-K, K-means, and Wang et al. Approaches

We examined the critical nature of load balancing in situations where servers have limited processing capacity. To do this, we fixed the queue size on each server to 150 tasks, and when a server's queue is full, the next arriving task is dropped and must be retransmitted later. The retransmission of tasks from the vehicle to the server consumes additional energy and degrades the latency. To this purpose, minimizing the number of dropped tasks is critical for decreasing and conserving energy consumption on cars and edge servers. The percentage of dropped tasks for each approach is depicted in Figure 7 . The

results indicate that our approach can deliver the best performance with the lowest dropped tasks rate. As a result, our method is more energy-efficient than the other placement strategies. Figure 8 shows the additional energy consumption caused by the retransmission of the dropped tasks. Specifically, our method consumes 48% less energy than Random-Random and 69% less energy than Random-Nearest. The Top-K strategy eliminates 5,67% of tasks, resulting in a 95,5% increase in energy consumption when compared to our solution. The K-means and Wang et al. approaches consume significantly more energy and increase by up to 394% compared to our solution.

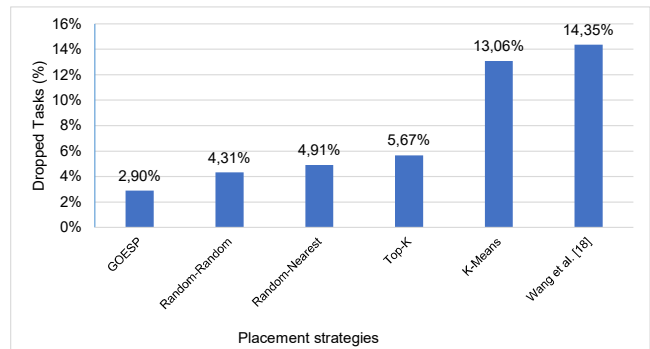


Figure 7. Dropped tasks results

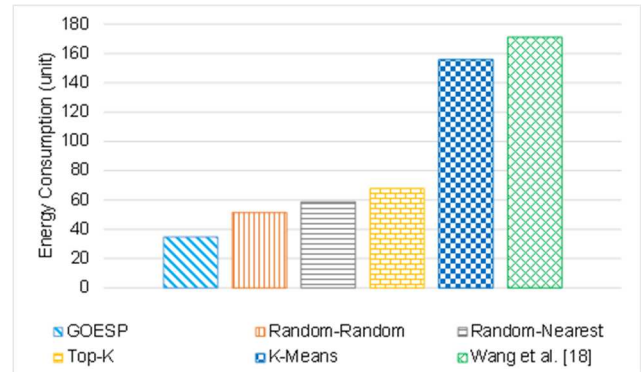


Figure 8. Additional energy consumption results

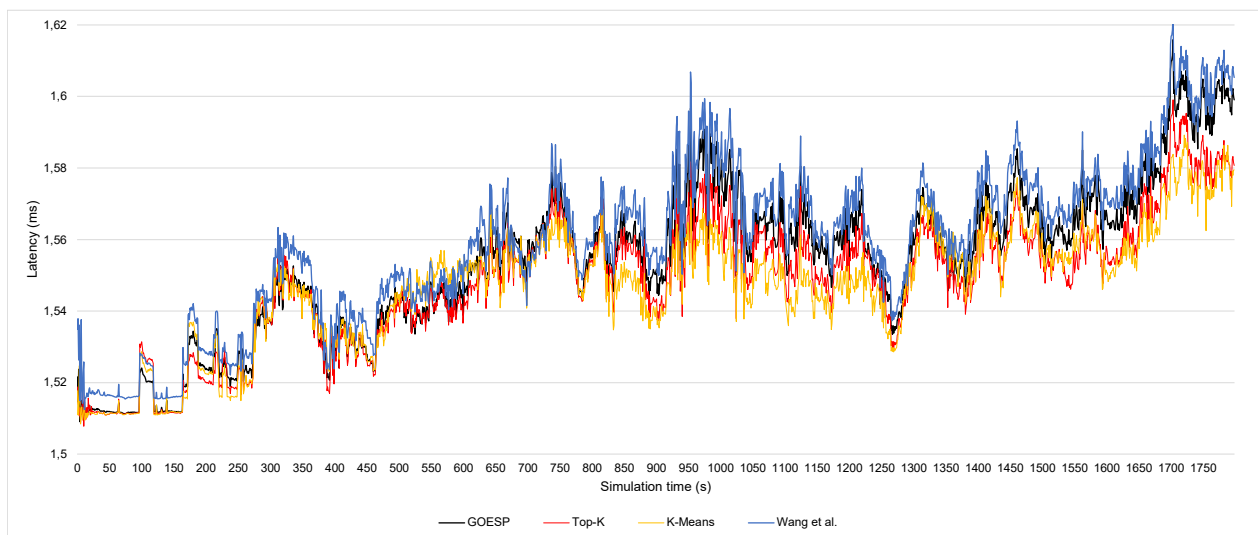


Figure 9: Latency simulation results: GOESP, Top-K, K-Means and Wang et al. [18].

Finally, we examine the effect of our solution on the average latency of messages received by the servers. Figure 9 depicts the latency curves generated using the simulation tool within the Artery framework for the approaches mentioned above. Our approach GOESP achieves a latency that varies between 1,51ms in low density and 1,61ms in high density. The other approaches offer a latency in approximately the same interval. When we examine the findings more closely, we see that GOESP outperforms Wang et al. [18]

Additionally, our solution presents a higher latency of 0,01ms (maximum) than the Top-K and K-means approaches. This increase could be explained by the fact that those approaches use the nearest server criteria, i.e., each RSU is connected to the nearest server; however, they do not consider other constraints like the server capacity and the workload balancing. For example, in the Top-K approach, server 2, with a capacity of 150 units, cannot handle all the roadside units' tasks (160 units of total demands). Therefore, this minor latency increase is acceptable for our solution to ensure that the server's capacity is not exceeded. On the other hand, the proposed model ensures that the latency threshold (L_{max}) constraint is always respected.

VI. CONCLUSION

Edge computing remains a new emerging technology that can boost the performance of various applications and help reduce power consumption by shifting the processing workload from the cloud to the edge. The energy-efficient Edge server placement problem has received little attention to the best of our knowledge. This paper investigated this topic in urban vehicle networks to optimize deployment costs, energy consumption, and balance workload between Edge servers while minimizing latency. We formulated the Edge server placement problem as an integer linear programming model. We used a methodology based on real-world Bordeaux traffic statistics to give a realistic Edge server placement. Our model considers Edge servers' capacities to balance servers' workloads efficiently. Using the AIMMS analytics platform, we proposed an effective placement solution. The obtained results indicate that our solution outperforms existing solutions in terms of energy efficiency. Workload balancing between servers prevents excessive energy consumption. Thus, our proposed placement technique balances deployment cost, energy consumption, workload balancing, and latency. As future work, we intend to investigate the service migration between Edge servers to ensure service continuity, decrease application delay, and further optimize energy consumption.

REFERENCES

- [1] Velez, Gorka, et al. "5g beyond 3gpp release 15 for connected automated mobility in cross-border contexts." *Sensors* 20.22 (2020): 6622.
- [2] Kocić, Jelena, Nenad Jovičić, and Vujo Drndarević. "Sensors and sensor fusion in autonomous vehicles." 2018 26th Telecommunications Forum (TELFOR). IEEE, 2018.
- [3] Buda, Su, et al. "Collaborative vehicular edge computing towards greener ITS." *IEEE Access* 8 (2020): 63935-63944.
- [4] Ning, Zhaolong, et al. "Mobile edge computing-enabled Internet of vehicles: Toward energy-efficient scheduling." *IEEE Network* 33.5 (2019): 198-205.
- [5] Habibullah, Khan Mohammad, Eric Rondeau, and Jean-Philippe Georges. "Reducing energy consumption of network infrastructure using spectral approach." *Technology for smart futures*. Springer, Cham, 2018. 235-250.
- [6] OpenDataBordeaux, Bordeaux Métropole, <https://opendata.bordeaux-metropole.fr/> [Accessed: May 13, 2022].
- [7] Zakaria, O., Britt, J., & Forood, H. U.S. Patent No. 9,717,012. Washington, DC: U.S. Patent and Trademark Office (2017).
- [8] SINGH, Pranav Kumar, NANDI, Sunit Kumar, et NANDI, Sukumar. A tutorial survey on vehicular communication state of the art, and future research directions. *Vehicular Communications*, 2019, vol. 18, p. 100164.
- [9] FESTAG, Andreas. Cooperative intelligent transport systems standards in Europe. *IEEE communications magazine*, 2014, vol. 52, no 12, p. 166-172.
- [10] PORAMBAGE, Pawani, OKWUIBE, Jude, LIYANAGE, Madhusanka, et al. Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, 2018, vol. 20, no 4, p. 2961-2991.
- [11] DATTA, Soumya Kanti, HAERRI, Jerome, BONNET, Christian, et al. Vehicles as connected resources: Opportunities and challenges for the future. *IEEE Vehicular Technology Magazine*, 2017.
- [12] HU, Yun Chao, PATEL, Milan, SABELLA, Dario, et al. Mobile edge computing—A key technology towards 5G. ETSI white paper, 2015, vol. 11, no 11, p. 1-16.
- [13] MACH, Pavel et BECVAR, Zdenek. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 2017, vol. 19, no 3, p. 1628-1656.
- [14] Zhang, Rui, et al. "An optimal roadside unit placement method for vanet localization." *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017.
- [15] Huang, P. C., Chin, T. L., & Chuang, T. Y. (2021). Server Placement and Task Allocation for Load Balancing in Edge-Computing Networks. *IEEE Access*, 9, 138200-138208.
- [16] Premsankar, Gopika, et al. "Efficient placement of edge computing devices for vehicular applications in smart cities." *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018.
- [17] Cao, Kun, et al. "Exploring placement of heterogeneous edge servers for response time minimization in mobile edge-cloud computing." *IEEE Transactions on Industrial Informatics* 17.1 (2020): 494-503.
- [18] Wang, Shangguang, et al. "Edge server placement in mobile edge computing." *Journal of Parallel and Distributed Computing* 127 (2019): 160-168.
- [19] Zhang, Ke, et al. "Predictive offloading in cloud-driven vehicles: Using mobile-edge computing for a promising network paradigm." *IEEE Vehicular Technology Magazine* 12.2 (2017).
- [20] Zhang, Ke, et al. "Mobile edge computing and networking for green and low-latency Internet of Things." *IEEE Communications Magazine* 56.5 (2018): 39-45.
- [21] Zhou, Zhenyun, et al. "BEGIN: Big data enabled energy-efficient vehicular edge computing." *IEEE Communications Magazine* 56.12 (2018): 82-89.
- [22] Ning, Zhaolong, et al. "Mobile edge computing-enabled Internet of vehicles: Toward energy-efficient scheduling." *IEEE Network* 33.5 (2019): 198-205.
- [23] Open Street Map France, "<https://www.openstreetmap.fr/>" [Accessed: May 14, 2022].
- [24] AIMMS, "Download AIMMS developer", 2020. [Online]. Available: <https://www.aimms.com/english/developers/downloads/download-aimms/>. [Accessed: May 13, 2022].
- [25] Varga, Andras. "Modeling and Tools for Network Simulation, Cap. OMNeT++." (2010).
- [26] Riebl, Raphael, Christina Obermaier, and Hendrik-Jörn Günther. "Artery: Large scale simulation environment for its applications." *Recent Advances in Network Simulation*. Springer, Cham, 2019. 365-406.
- [27] Lopez, Pablo Alvarez, et al. "Microscopic traffic simulation using sumo." 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018.
- [28] Boban, Mate, Joao Barros, and Ozan K. Tonguz. "Geometry-based vehicle-to-vehicle channel modeling for large-scale simulation." *IEEE Transactions on Vehicular Technology* 63.9 (2014): 4146-4164.