IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 May 2024, accepted 10 July 2024, date of publication 19 July 2024, date of current version 31 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3430826

RESEARCH ARTICLE

SLA-Based Service Provisioning Optimization in Vehicular Cloud Networks Using Fuzzy Logic

FARHOUD JAFARI KALEIBAR AND MARC ST-HILAIRE, (Senior Member, IEEE)
School of Information Technology, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Farhoud Jafari Kaleibar (farhoudjafarikaleiba@cunet.carleton.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2019-06263.

ABSTRACT Vehicular Cloud Networks (VCNs) enable vehicles to act as servers and share their abundant computing and storage resources. However, resource allocation in VCNs faces challenges due to factors like service pricing, resource variability, and mobility. This paper proposes a comprehensive approach for service provisioning in VCNs to address key challenges of quality of service, availability, and fair pricing. First, a mathematical model that considers service provider mobility, data volume, delay, cost, and location suitability is formulated. A high-level controller oversees network-wide service management by using fuzzy logic and calculating fit factors between requests and providers. Finally, a tailored heuristic algorithm is proposed to solve the NP-hard optimization problem efficiently. Simulations demonstrate the approach's effectiveness in maximizing allocation suitability under realistic VCN conditions.

INDEX TERMS Vehicular network, cloud services, service level agreement (SLA), service provisioning.

I. INTRODUCTION

In recent years, Vehicular Ad-hoc NETworks (VANETs) have garnered significant attention in the field of computer networks. These networks are made up of vehicles connected by wireless links and provide services such as traffic management and transportation by utilizing information and communication technologies [1]. With the increase in Internet of Things (IoT) applications and the incorporation of advanced sensors in vehicles, various applications have emerged, including vehicle health and safety, highway congestion management, and entertainment through the use of existing sensors. The processing of the generated data must be done quickly, which has led to the development of Vehicular Cloud Networks that provide bandwidth, storage, and processing services to users through the application [2].

The concept of Vehicular Cloud Networks (VCN) has had a positive impact on consolidating computing resources and improving drivers' situational awareness, thereby facilitating the transportation industry. However, the ultimate goal of VCN is to provide demand-based solutions to unpredictable events. The VCN can be dynamically adjusted according to the application requirements and the system environment. Vehicles typically have limited resources such as memory, computing power, and bandwidth due to the need for a small, low-cost hardware system. Conversely, many emerging applications require complex computing and extensive storage space, such as multimedia entertainment, social networks, and location-based services. One of the most effective solutions to address the problem of resource constraints is to share resources such as memory and computing among all vehicles or nearby infrastructures as a cloud [3]. One of the most significant challenges in providing services for vehicles is the optimal allocation of resources. In VCN, both the number of requests and the number of service providers are unknown and dependent on environmental conditions. Another challenge of providing VCN services is the temporary nature of the cloud formed due to the mobility of vehicles. Despite urban traffic and congestion in certain areas, the number of service requests and providers may increase, making the optimal selection of resources even more critical. Given these challenges, this problem has been shown to be NP-hard [4].

In this paper, we propose a novel mathematical model for addressing the resource provisioning challenge in vehicular clouds. The model takes into account various criteria to ensure compliance with Service Level Agreements

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchun Chen.

(SLAs), including service provider and receiver mobility, data volume, data transmission delay, and service price. Notably, the pricing criterion directly influences the problem formulation, considering the issue of a fixed price. This approach stands out from previous research by comprehensively considering multiple criteria and incorporating a pricing component in the problem formulation. The criteria used in this model offer a novel perspective on assessing the compatibility between the service and the service provider.

The paper also makes the following contributions:

- A high-level control strategy is employed to manage the services in the network. This strategy involves analyzing information obtained from the directory of each Roadside Unit (RSU) and considering various parameters to make comprehensive management decisions. However, despite the presence of this controller, all resources in the network are planned and managed in a centralized manner.
- A fuzzy logic-based approach used to evaluate the compatibility between service requesters and providers, considering diverse SLA criteria, including service provider and requester mobility, data volume, and transmission delay. This approach aims to determine the degree of suitability between the involved parties by leveraging fuzzy logic techniques.
- Finally, a heuristic algorithm has been proposed that is tailored to the conditions of the model, and customized for vehicular cloud networks.

The rest of the paper is organized as follows: In Section II, we provide a review of the related work on resource provisioning in vehicular cloud networks. The problem formulation and details of our proposed approach are described in Section III. Section IV presents the evaluation of our approach. Finally, in Section V, we conclude the paper.

II. RELATED WORK

This section reviews some of the most important works related to resource management in VCN. The paper [5] proposes an SDN-based task offloading architecture in fiber-wireless (FiWi) enhanced Vehicular Edge Computing Networks (VECNs) to minimize the processing delay of vehicles' computation tasks. It formulates the delay minimization problem and proposes three offloading schemes: two game theory-based algorithms called GTNOA and PGTOA, and an approximate load balancing algorithm called ALBOA. The paper [6] proposes a fuzzy-based method using a cuckoo search algorithm for energy-aware resource allocation in vehicular cloud computing. It aims to reduce energy consumption, SLA violation, execution time and response time. The fuzzy logic handles uncertainty and the cuckoo search algorithm optimizes resource allocation. The researchers in [7] proposes a new system called Vehicle as a Computational Resource (VaCR) that allows connected vehicles to share their unused computational resources within smart cities. Performance evaluation and quality of experience models are developed to classify vehicles based on their capabilities and satisfaction level. A multi-agent system and game theory model are used to optimize quality of experience for connected vehicles during resource provisioning. Extensive simulations show the VaCR system improves metrics like cost, classification, and time. The models are effective in optimizing quality of experience through the exploitation of distributed vehicle computing power. The paper [8] proposes a dynamic service migration algorithm for vehicular clouds. It aims to efficiently map user requests to virtual machines hosted on vehicles. The algorithm considers three vehicle types and performs partial request assignment and migration. Extensive simulations show it outperforms other algorithms in metrics like completed requests and migration rate.

Some of the other works have focused on optimization issues. For example, paper [9] proposes a multi-objective optimization model for resource allocation in vehicular cloud computing networks. The goals are to minimize blocking probability and provider cost. Constraints include vehicular cloud characteristics like connection duration and request deadlines. To solve this NP-hard problem, an improved Non-dominated Sorting Genetic Algorithm (NSGA-II) called AC-INSGA is developed, which modifies the initial population based on a matching factor and uses dynamic crossover and mutation probabilities. The authors in [10] study optimal edge cloud resource provisioning for connected vehicle fleets to minimize cost while ensuring quality of service. It models vehicle mobility uncertainty using arrival and departure time distributions. An optimization model is proposed to minimize provisioning cost per cell with a constraint on blocking probability below a threshold. A two-phase algorithm using bracketing and binary search solves the problem. In [11], the authors propose resource pooling in vehicular fog computing where vehicles contribute their computing resources to a community sponsored by a roadside unit. The goal is to maximize the benefits of vehicles by optimally selecting which community to join. A genetic algorithm is developed for the decision making. The utility function accounts for dwell time, available resources, pricing, competitor strategies, and request rate. While prior works in VCN resource management have addressed optimization of parameters like latency, cost and resource pooling, they lack a comprehensive approach that considers multiple quality of service criteria together under dynamic network conditions. Most existing studies also do not incorporate pricing as a key factor in the resource allocation problem formulation. This can lead to issues with maintaining equitable costs for providers and requesters over time. In addition, the proposed approach aims to find a suitable matching between requesters and providers by evaluating multiple parameters from both sides, unlike other works that do not comprehensively

TABLE 1. Comparison of related work.

Related Work	Supported Cloud Type	Service Quality Attributes	Overall Method
[11]	Vehicular cloud, RSU cloud	Dwell time, pricing, competitor strategies, request rate	Formulated as optimization problem, solved using genetic algorithm
[10]	Vehicular cloud, RSU cloud, Conventional cloud	Service blocking probability	Optimization model with constraints, bracketing and binary search algorithm
[9]	Vehicular cloud	Blocking probability, cost	Multi-objective optimization model, improved NSGA-II algorithm
[8]	Vehicular cloud, Conventional cloud	Service continuity	dynamic service migration algorithm for vehicular clouds
[7]	Vehicular cloud	Percentage of satisfaction (cost, security, performance) and device capability	Optimizing QoE using an ML algorithm
[6]	Vehicular cloud, RSU cloud	Efficiency	A new fuzzy based method with Cuckoo search algorithm
[5]	Vehicular cloud, RSU cloud, Conventional cloud	Delay, cost	Designed task offloading scheme based on cost and delay
Proposed Approach	Vehicular cloud, RSU cloud, Conventional cloud	Delay, cost, volume, mobility management	Multi-objective optimization model, new heuristic to solve the problem

consider factors impacting both parties. Table 1 provides a summary of the related work.

III. PROBLEM FORMULATION

A. OVERVIEW

This section proposes an approach that utilizes service level agreements for service provisioning in vehicular cloud networks. The primary objectives of this approach are to provide quality services to requesters while maintaining performance, ensuring fixed costs and increasing service availability, regardless of the mobility of service providers and requesters. Table 2 shows the notation used in the article.

Service management typically consists of two parts: service discovery and advertising, and service provisioning and delivery. For service discovery and advertising, we employ the approach presented in [12], with slight modifications to support the Trusted Third Party (TTP) component. Similar to [12], this paper uses a 3-layer cloud architecture to manage the service. Starting from this point in the article, it is important to clarify that the term VCN will encompass all three layers of the cloud.

The controller (C_H) is situated in the central cloud, which manages the top layer cloud resources, RSUs and their available resource. Additionally, a TTP broker hosts the service and provides a service level agreement [13].

As illustrated in Fig. 1, the proposed architecture utilizes a TTP that manages service provisioning in a hybrid manner. The TTP oversees both distributed service directories located at each RSU, where provider/requester specifications are registered locally. It also centrally orchestrates service parameters as the coordinator working with the hierarchical controller structure. This hybrid management approach enables local optimization through decentralized directories while benefiting from centralized coordination of network-wide service information and dynamic conditions. The coordinated efforts between distributed directories, centralized controller and the TTP as the orchestrator facilitate optimal matching of requests to providers under unpredictable VCN environments [12]. Vehicles periodically send service requests/messages to the nearest RSU, which are recorded by controller (C_H). If a vehicle moves from one RSU area to another within a fixed controller zone, the movement is detected and recorded by the controller when registering the message in the new RSU. Moreover, each TTP can locally manage its resources under an RSU and manage resources on the controller by planning for requester/provider mobility. With such a comprehensive view of resources and services, it will be possible to create a suitable pricing framework.

Service providers register their service specifications at the nearest RSU (RSU service directory), including the type of resource/service, data volume, cost per unit, service duration and current location. On the other hand, service requester vehicles send their requested service specifications, such as type of service, desired data volume, service duration and current location to their nearest RSU.

The primary focus of this research is the second part of service management, namely the provisioning and delivery of services while maintaining quality of service. The first step is to identify the right quality of service and service provider for a request by assessing the features of the request and service provider. After planning the appropriate service

TABLE 2. Variable descriptions.

Variable	Description
N	Set of requested services where i indicates a specific request ($i \in$ N)
M	Set of service providers where j specifies a specific service provider ($j \in$ M)
X_{ij}	Binary variable that indicates whether service provider j is selected for service request i ($x_{ij}=1$) or not ($x_{ij}=0$)
s_{ij}	Degree to which service provider j's speed affects service i
v_{ij}	Suitability of provider j's storage capacity for the service i
c_{ij}	Suitability of provider j's cost for service i (compared to the cost of a similar service for other providers), whose values are determined based on equations 2 and 3
d_{ij}	Suitability of service provider j's delay for service i
l_{ij}	Geographical distance (location) between service provider (vehicle) j and service requester i, whose values are determined based on Table 5
$Cprice_{ij}$	Comparative pricing of provider j for service i
$Cprice'_{ij}$	Previous or initial comparative price of service i by provider j
P_{ij}	Price of service i provided by j
$\max_k(P_{ik})$	Maximum price of service i which is provided by k
MS_i	Mobility impact on service i
MP_j	Mobility of provider j
SDV_i	Required data volume for the service i
PSC_j	Provider j's storage capacity
SDT_i	Service i's delay tolerance
PDD_j	Provider j's delivery delay
R/P	Ratio of requesters to the providers.
$\alpha, \beta, \gamma, \delta, \epsilon$	Coefficients showing the relative importance of each suitability factor

for the requester, the service is provided under a service quality agreement. Nonetheless, there are still two important concerns for the requester that the TTP is responsible for resolving. The first concern is the service availability guarantee, which may be violated due to vehicle mobility, and the second concern is the service price guarantee. The concept of a service price guarantee ensures that the pricing of a service is equitable for both the provider and the recipient. This entails maintaining a fair service price throughout the entire service delivery process, even if there are changes in the resources being utilized.

B. ASSUMPTIONS

The following assumptions are made in the development of the proposed model.

- The locations of RSU nodes are static and predetermined.
- Each service request can be assigned to a vehicular node, RSU, or the conventional cloud.
- The bandwidth between a requester and its provider node is plentiful to fulfill the requested service.

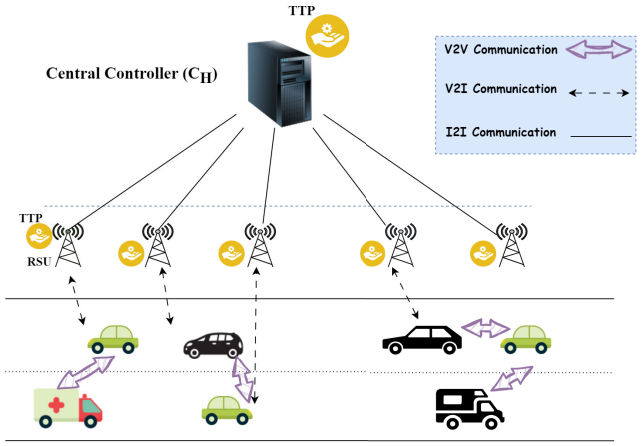

FIGURE 1. Proposed architecture for VCN.

- The conventional cloud is situated at a considerable distance from the users when compared to the fog nodes (vehicular and RSU clouds).
- The V2V and the V2I connectivity are established using the DSRC protocol and the connectivity between the controllers and the RSUs is wired (optical fiber).

C. FUZZY LOGIC-BASED SUITABILITY CALCULATION

Due to the involvement of numerous environmental variables, the correspondence between the nature of a service and its provider is not easily reducible to a mathematical formulation. Therefore, fuzzy logic has been utilized to compute appropriate values for the metrics defined in the optimization formula presented in this section.

Fuzzy logic is a decision-making process that operates using input membership functions and a set of fuzzy rules. Additionally, this approach has demonstrated great efficiency in real-time systems [14]. To design a fuzzy inference system, the first step is to identify input and output variables and their corresponding fuzzy membership functions. Next, fuzzy rules are created to represent the knowledge base of the inference engine. The final step involves calculating the fit factor using defuzzification [15]. We propose a new Fuzzy Logic Controller (FLC) for the calculation of the fit factor value which is assumed to be deployed in C_H. The procedure of calculating fit factor based on fuzzy logic has three steps as described in the following.

1) METRIC COMPUTATION AND FUZZIFICATION

In this section, three distinct fit factors should be calculated for the service delay suitability, mobility impact suitability, and data volume suitability. The mobility impact (s_{ij}) is the first parameter considered for evaluating the suitability of a service provider for a given service. To determine the impact of mobility on the service (MS_i), we use the time required to provide the service as a metric, which varies depending on the type of application [16]. For instance, safety-driving applications require only a few seconds to

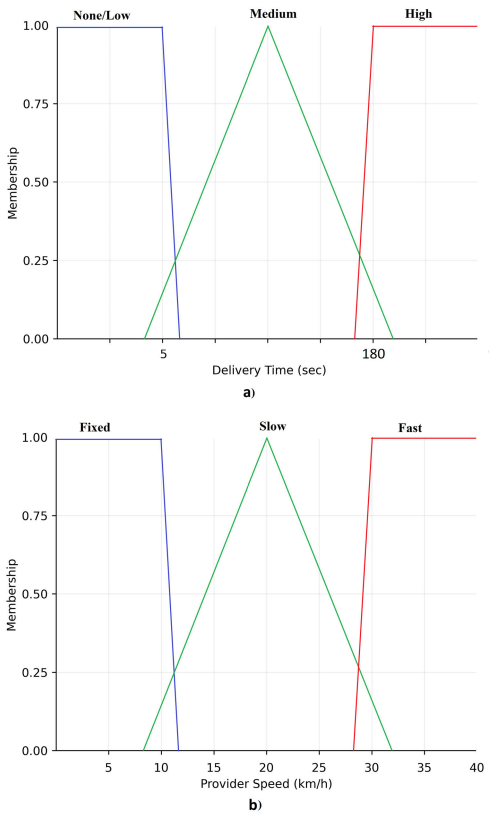

FIGURE 2. Fuzzification of a) Mobility Impact on Service b) Provider Mobility.

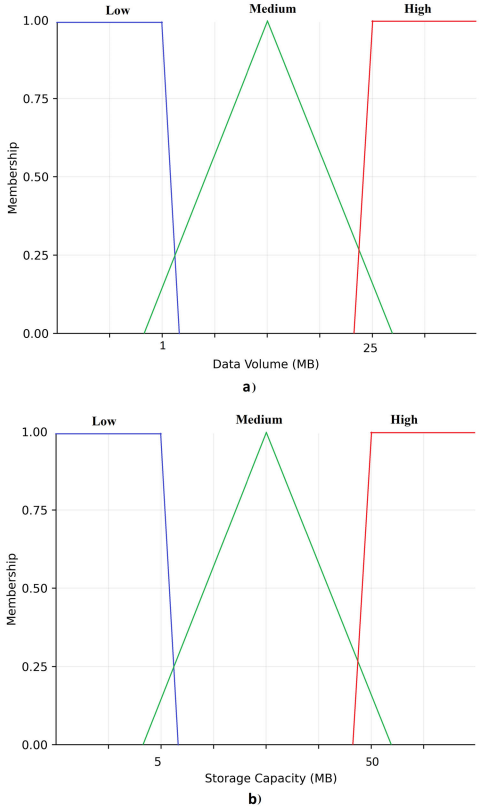

FIGURE 3. Fuzzification of a) Required Data Volume for the Service b) Provider Storage Capacity.

exchange messages [17], while other applications, such as advertising and games, take several minutes. Streaming applications, on the other hand, may require much longer periods [18]. For the provider side (MP_j), the speed of vehicles on the street is considered, where speeds exceeding 30 km/h indicate a fast provider, while speeds below 10 km/h indicate a slow and fix provider. Fig. 2 illustrates the fuzzification stage of this metric.

The second parameter considered is the data volume suitability (v_{ij}), which evaluates how well a potential service provider's storage capacity (PSC_j) matches the amount of data required to provide the requested service (SDV_i). Drawing from the simulation assumptions detailed in [19], it is posited that a data threshold of 1 megabyte signifies a low volume for services like safety driving. Conversely, a data threshold of 30 megabytes or higher is regarded as significant, for an applications like video streaming [20]. We also assume that memory-constrained vehicles have a memory capacity of around 5 megabytes, whereas other vehicles (and other non-vehicle providers) have a memory capacity of over 50 megabytes. Fig. 3 illustrates the fuzzification of this parameter using similar reasoning.

The third parameter considered is the delay tolerance (d_{ij}) suitability between the service request (SDT_i) and potential providers (PDD_j). For the final parameter of fuzzification, we use the values specified in [21] to determine the service delay tolerance. If a service tolerates a delay of less than 0.1 seconds, it is highly sensitive, while a delay of 0.1 to 0.7 seconds is considered medium. If the service tolerate delay exceeds 0.7 seconds, the service is considered insensitive to delay. Additionally, the service provider will have a similar level of delay based on the above parameters (see Fig. 4).

2) RULES MAPPING

Based on the fuzzy values of MP_j and MS_i, SDV_i and PSC_j and finally SDT_i and PDD_j, the inference engine maps the fuzzy values to the IF-THEN rules contained in the knowledge Rule Base and defined in Tables 2 to 5 as the output fuzzy value. The linguistic variables of the output are defined as Perfect, Good, Acceptable, Bad, Very Bad and described as the following:

- **IF** MS_i is {None/Low, Medium, High}, and MP_j is {Fixed, Slow, Fast}, **THEN** s_{ij} is {Perfect, Good, Acceptable, Bad, Very Bad}.
- **IF** SDV_i is {Low, Medium, High}, and PSC_j is {Low, Medium, High}, **THEN** v_{ij} is {Perfect, Good, Acceptable, Bad, Very Bad}.
- **IF** SDT_i is {Not Sensitive, Medium, Sensitive}, and PDT_j is {Low, Medium, High}, **THEN** d_{ij} is {Perfect, Good, Acceptable, Bad, Very Bad}.

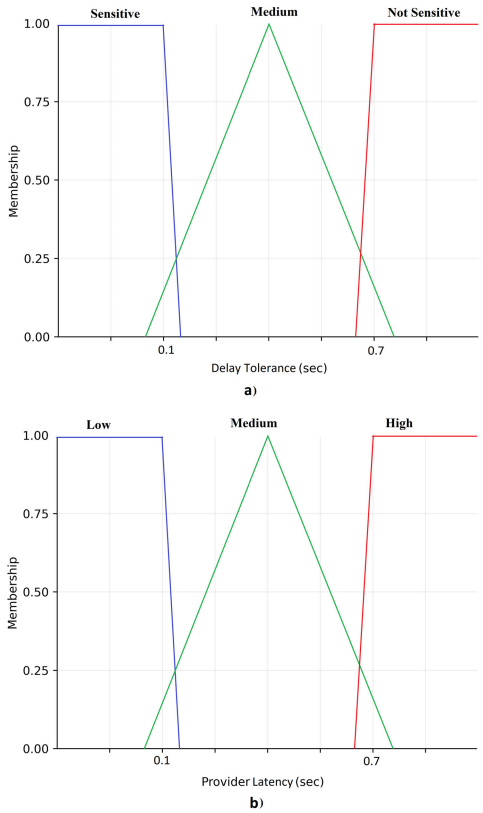

FIGURE 4. Fuzzification of a) Delay Tolerance of Service b) Provider Latency.

For Table 6, the same rule has been used, but since it is a single parameter, there is no need to calculate the related metric.

3) DEFUZZIFICATION

In this step, the defuzzifier takes the fuzzy output value and converts it to a crisp value using predefined output membership function as indicated in Fig. 5. The defuzzification process utilizes the center-of-gravity method, which is commonly used in [22]. For example, if the degrees of membership for the fuzzy sets Bad, Acceptable, and Good are 0.5, 0.75, and 0.5, respectively, the resulting function takes on a shape as shown in Fig. 5. The centroid of this shape is then calculated to obtain the defuzzified output.

D. MATHEMATICAL MODEL

The objective is to maximize the total suitability of the selected service providers for a set of service requests in a vehicular cloud network. The suitability is determined based on several factors, including cost, delay, speed, service volume, and geographical distance. Therefore, we will face an optimization problem with the objective of maximizing the allocation of services to the requesters while considering

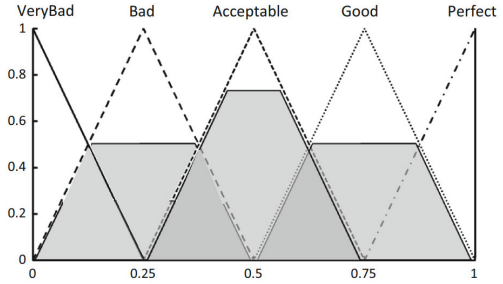

FIGURE 5. Defuzzification output.

certain conditions. The objective function is:

$$Maximize \sum_{i}^{n} \sum_{j}^{m} (X_{ij} \cdot (\alpha \cdot v_{ij} + \beta \cdot s_{ij} + \gamma \cdot c_{ij} + \delta \cdot d_{ij} + \epsilon \cdot l_{ij})) \tag{1}$$

Subject to:

- Each service can only be assigned to one service provider:

$$\sum_{j}^{m} X_{ij} \leq 1 \quad \text{for all } i \in \{1, \ldots, N\}.$$

- Each service provider can only be assigned to one service within a certain duration

$$\sum_{i}^{n} X_{ij} \leq 1 \quad \text{for all } j \in \{1, \ldots, M\}.$$

- Binary variable constraint:

$$X_{ij} \in [0, 1] \quad \text{for all } i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}.$$

- The sum of the coefficients for the first set of parameters will be equal to 1:

$$\alpha + \beta + \gamma + \delta + \epsilon = 1$$

- The values of the suitability parameters range between 0 and 1:

$$v_{ij}, s_{ij}, c_{ij}, d_{ij}, l_{ij} \in [0, 1]$$

- To calculate c_{ij} we have:

$$c_{ij} = \begin{cases} 1 - Cprice_{ij} & \text{if } Cprice_{ij} \leq Cprice'_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$Cprice_{ij} = \frac{P_{ij}}{\max_k(P_{ik})} \tag{3}$$

The objective function is a weighted sum of the suitability values for each selected service provider. The weights are determined by the coefficients α, β, γ, δ, and ϵ, which represent the relative importance of each suitability factor. The objective is to maximize the total suitability of selected service providers for all services. The constraints ensure that each service is assigned to only one service provider and that

TABLE 3. Mobility impact for service provider mobility.

Mobility Impact for Service	Provider Mobility	s_{ij}
None/low	Fixed	Acceptable
None/low	Slow	Good
None/low	Fast	Perfect
Medium	Fixed	Good
Medium	Slow	Acceptable
Medium	Fast	Bad
High	Fixed	Perfect
High	Slow	acceptable
High	Fast	Very Bad

TABLE 4. Service data volume requirement.

Service Data Volume Requirement	Provider Capacity	v_{ij}
low	Low	Perfect
low	Medium	Good
low	High	Acceptable
Medium	Low	Very Bad
Medium	Medium	Perfect
Medium	High	Good
High	Low	Very Bad
High	Medium	Bad
High	High	Perfect

TABLE 5. Service delay tolerance.

Service Delay Tolerance	Provider Delay	d_{ij}
Not Sensitive	Low	Bad
Not Sensitive	Medium	Acceptable
Not Sensitive	High	Perfect
Medium	Low	Good
Medium	Medium	Perfect
Medium	High	Bad
Delay Sensitive	Low	Perfect
Delay Sensitive	Medium	Acceptable
Delay Sensitive	High	Very Bad

each service provider is assigned to only one service within a certain duration. The binary variable constraint ensures that service providers are either selected ($X_{ij} = 1$) or not selected ($X_{ij} = 0$).

In conclusion, this optimization problem can be used to efficiently select service providers for a set of services in a vehicular cloud network based on various suitability factors. The objective is to maximize the overall suitability of selected service providers while ensuring that each service is assigned to only one service provider and that each service provider is assigned to only one service within a certain duration.

E. PROPOSED ALGORITHM TO SOLVE THE PROBLEM

Given the NP-hard nature of the problem, network simulators can't solve large problem instances within acceptable time limits. Therefore, a heuristic approach was developed and evaluated.

The heuristic algorithm sorts all request-provider tuples based on their respective scores (R_{ij}), calculated using Eq. 4. The matrix $\mathbf{R}$ contains all the calculated scores R_{ij}, as shown

TABLE 6. Service requester and provider locations.

Service Requester and Provider Locations	l_{ij}
Same RSUs	Perfect
Different RSUs but Same C_1	Good
Different C_1 but in the neighborhood	Acceptable
Else	Very Bad

in Eq. 5.

$$R_{ij} = (\alpha \cdot v_{ij} + \beta \cdot s_{ij} + \gamma \cdot c_{ij} + \delta \cdot d_{ij} + \epsilon \cdot l_{ij}) \tag{4}$$

$$\mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1j} \\ R_{21} & R_{22} & \cdots & R_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ R_{i1} & R_{i2} & \cdots & R_{ij} \end{pmatrix} \tag{5}$$

The heuristic seeks to find the pair of tuples whose sum of scores is maximum among all other pairs, while ensuring that the selected tuples do not share any common providers or requesters. This is represented by Eq. 6.

$$\mathbf{S_t} = \{(R_{ij}, R_{kl}) \mid i \neq k, j \neq l, \\ R_{ij} + R_{kl} \geq \sum_{i'i'',j'j''}^{n,m} (R_{i'j'} + R_{i''j''}),\ i' \neq i'', j' \neq j''\} \tag{6}$$

Once a pair of tuples is selected, it is removed from further consideration and added to the final set of selected Tuples (S). This iterative process continues until no tuples remain. The final value of S will be the solution, as shown in Eq. 7. The proposed algorithm is outlined in Algorithm 1.

$$\mathbf{S} = \bigcup_{t=1}^{M} \mathbf{S_t} \tag{7}$$

IV. SIMULATION

A. SIMULATION ENVIRONMENT

This section presents the results of the simulations that were performed to evaluate the proposed approach. For this purpose, we used the Network Simulator 2.35 software (with the 802.11p amendment and the Nakagami propagation model) [23]. Using the ns2 simulation tool, we implemented a scenario where vehicles were represented as mobile nodes, and RSUs (Roadside Units) were represented as conventional nodes. As illustrated in Fig. 6, we employed the C++ environment of ns2 to implement the problem formulation and solver algorithm. Additionally, by utilizing the Tcl environment, we successfully implemented various scenarios by adjusting parameters such as the number of vehicles, their speeds, and randomly generating requests. This approach facilitated the simulation and analysis of the system's behavior under diverse conditions, enabling us to gain valuable insights into the network's performance and efficiency. Each simulation scenario was repeated a minimum of 20 times, with some scenarios undergoing additional runs for further analysis, and the final results are the average of these runs. In each execution of the scenarios, the nodes

Algorithm 1 Heuristic Solver Algorithm

Input: R (Matrix of requests/providers scores)
Output: S (set of selected request/provider tuples)
$\mathbf{S} \leftarrow \emptyset$
continue $\leftarrow$ true
while *continue* **do**
  continue $\leftarrow$ false
  for $i \leftarrow 1 to \text{size}(N)$ **do**
    for $j \leftarrow 1 to \text{size}(M)$ **do**
      for $k \leftarrow 1 to \text{size}(N)$ **do**
        for $l \leftarrow 1 to \text{size}(M)$ **do**
          Select tuple R_{ij} in R
          Select tuple R_{kl} in R
          if $i \neq k\ j \neq l$ **then**
            if $R_{ij} + R_{kl}$ is maximum among selected tuples in $\mathbf{S} and R_{ij}$ and R_{kl} do not share any common provider/requester **then**
              $\mathbf{S} \leftarrow \mathbf{S} \cup R_{ij}, R_{kl}$
              Remove R_{ij} and R_{kl} from **R**
              continue $\leftarrow$ true
            end if
          end if
        end for
      end for
    end for
  end for
end while
return final solution **S**

TABLE 7. Simulation parameters.

Parameters	Value	Default
Simulation time	100 sec	-
Simulation area	1.5 * 1.5 Km	-
Radio transmission range	200 m	-
Request/register packet size	100 bits	-
Data volume	1 to 50 MB	-
Number of vehicles	100 to 500	300
Vehicle average speed	10 to 60 Km/h	30 Km/h
Number of RSUs	20	-
R/P	1/2 to 4/2	3/2
$\alpha, \beta, \gamma, \delta, \epsilon$	0.1 to 0.6	0.2

(vehicles and RSUs), events (service requests and provider specifications), and vehicle mobility are randomly generated (RWP model). We also consider, for all scenarios, that the network congestion is set at 100%, implying that all vehicles in the network operate as either requesters or providers. Other simulation parameters are shown in Table 7. For each parameter, there is a default value that is utilized when the parameter remains fixed and does not vary during a specific evaluation. However, when the parameter is variable in a scenario, its value is explicitly specified.

We use the following metrics to evaluate the proposed approach:

- **Provisioning Score**: The provisioning score is defined as the sum of scores obtained from mapping suitable providers to requesters. It is calculated using Eq. 1. A higher provisioning score indicates that the algorithm performs more effectively.
- **Packet Delivery Rate (PDR)**: The packet delivery rate refers to the proportion of correctly received packets divided by the total number of packets sent (in percent).
- **SLA Adherence Rate**: SLA adherence rate is a metric similar to PDR, but with additional considerations for different types of applications. In the case of services with low tolerance for delay, if the packet delivery time exceeds the defined tolerance threshold, it is considered as an SLA violation. Such violations contribute to a reduction in the SLA adherence rate. Furthermore, for applications that involve high-volume transmission, if more than 5% of the transmitted packets fail to reach the destination correctly, the entire service (whole related packets) is considered dropped (95% of reliability).
- **Resource Utility**: Resource utility is a metric that quantifies the extent to which the capacity of a potential provider is utilized to fulfill service requests within a given time period. It is expressed as a percentage, indicating the proportion of the provider's capacity that is effectively utilized during that time period.
- **Service Delay**: the time from sending a request to receive the first part of the service (in milliseconds).

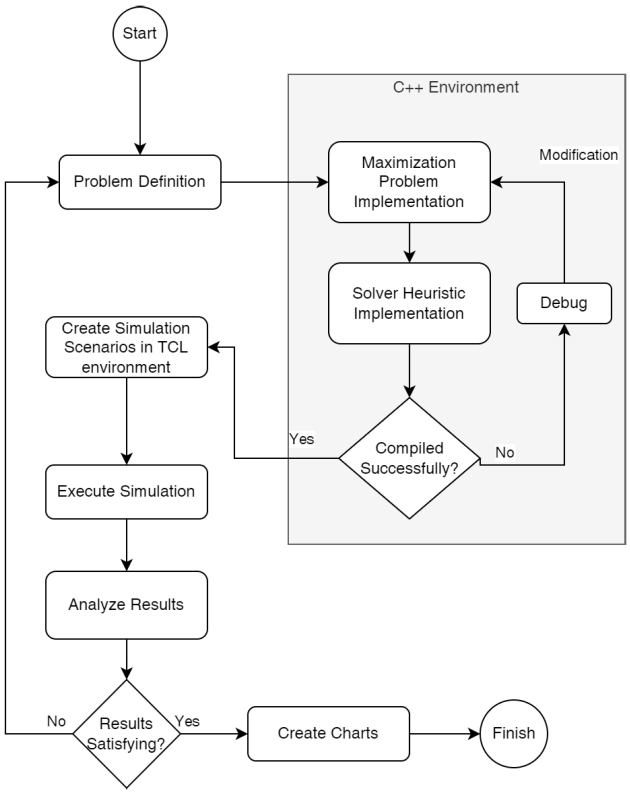

FIGURE 6. Flowchart depicting the implementation steps of the Proposed Method.

TABLE 8. Comparison of results in term of provisioning score: proposed method vs greedy algorithm vs CPLEX.

Provisioning Score			Performance (%)	
Proposed Method	Greedy Algorithm	CPLEX	Proposed Method	Greedy Algorithm
86.5	86.5	99.25	87.15	87.15
99	98.75	103.75	95.42	95.18
78.75	78.75	90.5	87.02	87.02
96.25	95.5	104.25	92.33	91.60
77.75	77.5	97.25	79.95	79.69
102	102	110.5	92.31	92.31
92.25	92.25	105.75	87.23	87.23
103.25	102.75	115	89.78	89.34
97.25	97.25	104.5	93.06	93.06
101.25	101.25	109.5	92.47	92.47
106	106	114.5	92.58	92.58
102.25	102	118.25	86.47	86.26
100.25	99	115.5	86.80	85.71
74	74	86.5	85.55	85.55
108.75	108.5	118.25	91.97	91.75
97.75	97.75	107.75	90.72	90.72
100.5	100.5	117.5	85.53	85.53
102.75	102.5	114.75	89.54	89.32
100	100	103.5	96.62	96.62
82	82	104.75	78.76	78.76
Average:			91.09	90.81

B. RESULTS AND DISCUSSIONS

In this section, the simulation results are examined and discussed in details. In the conducted simulations, the performance of the Proposed Method was compared to two other methods: the Greedy Algorithm and the SLA-based method [13]. The Greedy Algorithm operates by sorting tuples based on their score and consistently selecting the tuple with the highest score at each iteration. Subsequently, the algorithm removes both the requester and provider from the selected tuple, and the process is repeated. It should be noted that since the SLA-based method does not involve a maximization problem, the provisioning score is not relevant and therefore omitted in this context.

1) PROVISIONING SCORE

Initially, the service provisioning problem was formulated and solved optimally using the CPLEX solver to obtain an optimal solution. This served as a benchmark for evaluating the performance of our proposed heuristic. Subsequently, we implemented our heuristic algorithm in a simulator and conducted experiments in a network environment. The simulator allowed us to simulate the behavior of the heuristic algorithm under realistic conditions.

To assess the effectiveness of our heuristic approach, we compared the results obtained from running the heuristic algorithm in the simulator with the optimal solution obtained from the CPLEX solver with the same features and situations.

Table 8 presents the performance of the Proposed Method and Greedy Algorithm compared to the CPLEX solver. Performance is defined as the ratio of the score achieve by the respective methods over the score of the optimal solution (CPLEX). By comparing these performance values, we can evaluate how closely the Proposed Method approximate the optimal solution provided by the CPLEX solver. It should be noted that the results presented here are obtained from a series of experiments consisting of 60 requests and 40 providers (R/P=3/2) within a specific time period. The experiments were conducted for a total of 20 runs to ensure accuracy and reliability of the data.

When looking at the results from Table 8, the overall performance of the Proposed Method is around 91.09% (versus 90.81% for the Greedy Algorithm).The lower scores observed can be attributed to scenarios where the metrics have similar values, requiring the algorithms to make more complex calculations in order to select the most appropriate values. While the CPLEX solver achieves optimal results in such scenarios, its time-consuming nature makes it impractical for real-world networks like vehicular networks.

By adapting to the dynamic and complex nature of vehicular networks, the Proposed Method and the Greedy Algorithm aim to strike a balance between computational efficiency and achieving satisfactory results. Moreover, by comparing the performance of the Proposed Method with the Greedy Algorithm, it is evident that the Proposed Method never under-performs the Greedy Algorithm, and overall, the Proposed Method exhibits slightly better performance. This slight performance improvement can have a significant impact on other parameters, which are presented in the subsequent subsections.

2) VARYING NUMBER OF VEHICLES

In this section, we evaluate the Proposed Method by varying the number of vehicles. As we consider varying the number of vehicles within a fixed simulation area, the main objective is to assess how the algorithm performs and adapts to changes in network density and the corresponding impact on service demand and provider conditions. Additionally, it can also give us insights about the scalability of the proposed algorithm, determining if it can be effectively utilized in denser environments.

In Fig. 7a, as the number of vehicles increases, the network traffic and the number of service requests also increase. This leads to a higher demand for resource allocation and puts additional strain on the network, resulting in a decrease in the packet delivery rate. This phenomenon explains the decreasing slope observed in the PDR values across all three methods. Comparing the three methods, it is evident that both the Greedy Algorithm and the Proposed Method consistently outperform the SLA-based approach in terms of packet delivery rate. This difference can be attributed to the more restrictive parameters and formulas used in the SLA-based method, as well as its strict rules mapping table. These factors limit the flexibility in choosing providers and resource allocation, consequently affecting the packet delivery rate negatively.

In Fig. 7b, heightened network activity leads to increased delays and packet loss, resulting in a decrease in SLA adherence across all three methods. This explains the decreasing trend observed in the SLA adherence values. Comparing

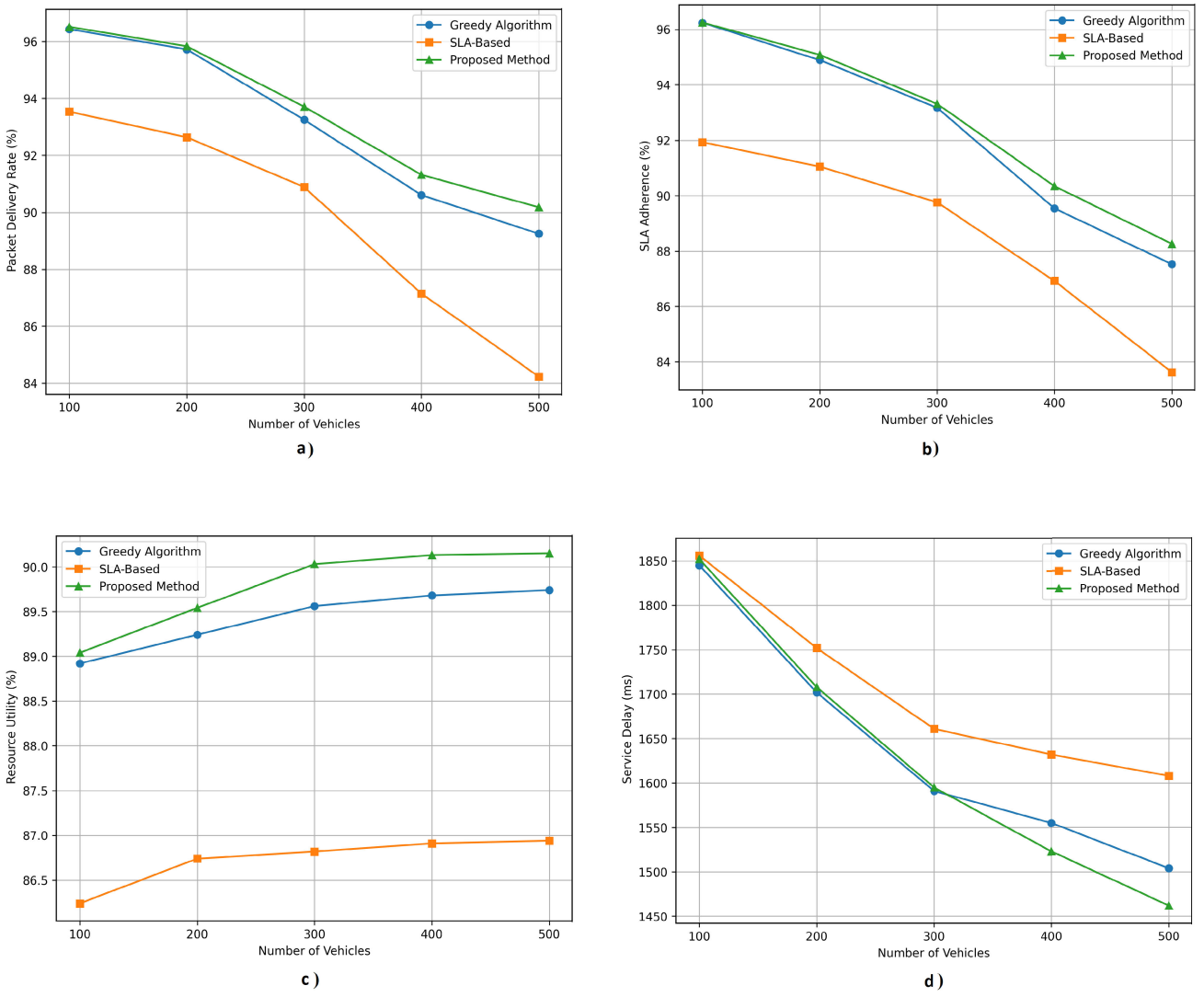

FIGURE 7. Performance comparison when varying the numbers of vehicles in terms of: a) PDR, b) SLA Adherence, c) Resource Utility, d) Service Delay.

the three methods, it is evident that both the Greedy Algorithm and the Proposed Method consistently outperform the SLA-based approach in terms of SLA adherence. This discrepancy can be attributed to the more flexible resource allocation strategies employed by the Greedy Algorithm and the Proposed Method, which allow for better adaptation to changing network conditions and service demands.

Fig. 7c, presents the resource utility, represented as percentages, for the Greedy Algorithm, the Proposed Method, and the SLA-based approach across different numbers of vehicles in a vehicular network scenario. Upon analyzing the chart, it is clear that both the Proposed Method and the Greedy Algorithm consistently outperform the SLA-based approach in terms of resource utility. One of the key reasons for this difference is that the SLA-based approach imposes limitations on the utilization of providers within its formula. The increasing slope observed in the resource utility scores as the number of vehicles increases can be attributed to the impact of a larger number of providers and service requests on the optimization of resource utility. With an increasing number of vehicles, the number of potential providers also tends to increase, resulting in a larger pool of available resources. The availability of ample resources presents a favorable opportunity to optimize their utilization effectively, aiming to maximize resource efficiency.

In Fig. 7d, as the number of vehicles increases, the service delay decreases for all three approaches. This reduction in service delay is attributed to the increased vehicle density, which leads to a decrease in transmission times due to the availability of more potential providers in closer proximity. Among the three approaches, the Proposed Method exhibits the lowest service delay across all vehicle densities, followed by the Greedy Algorithm and then the SLA-Based method. The superior performance of the Proposed Method and the

Greedy Algorithm can be attributed to their consideration of the locations of providers and requesters when selecting them for service provisioning. By selecting providers and requesters that are in close proximity, these methods can minimize the transmission delays, resulting in lower overall service delays compared to the SLA-Based method, which may not prioritize geographic proximity as a selection criterion. It is noteworthy that while the Proposed Method consistently outperforms the other two approaches, the performance gap between the Proposed Method and the Greedy Algorithm widens as the number of vehicles increases. This suggests that at higher vehicle densities, the advantage of the Proposed Method's more sophisticated selection criteria diminishes compared to the Greedy Algorithm, potentially because the increased availability of nearby providers makes the simpler selection approach employed by the Greedy Algorithm almost as effective.

In conclusion, the proposed method consistently outperformed the Greedy Algorithm and SLA-based method across the performance metrics considered, including packet delivery rate, SLA adherence, resource utility, and service delay. However, it is important to note that the performance gap between the Proposed Method and Greedy Algorithm narrowed in most charts. While the Greedy Algorithm exhibited comparable performance to the Proposed Method in metrics such as packet delivery rate, SLA adherence, and service delay, a significant difference existed in resource utility between the two approaches. As shown in the resource utility chart, the Proposed Method achieved substantially higher values compared to the Greedy Algorithm across all vehicle densities. This indicates that the Proposed Method is scalable and enhances the utilization of network resources, which can potentially result in improved provisioning capacity and overall efficiency.

3) VARYING AVERAGE SPEED OF VEHICLES

In this section, we varied the average speed of vehicles to analyze the performance of the compared methods. This provides important insights, as the average speed of vehicles typically differs based on factors such as road type, traffic conditions, and time of day. For example, vehicles tend to move faster on highways during non-peak hours compared to city streets during rush hour. Accounting for a range of maximum speeds helps evaluate how well each method can adapt to these real-world scenarios with differing vehicle mobility. By changing the vehicle speed distribution, we are essentially altering the density and available connectivity opportunities in the network at different time periods. This allows us to gauge the sensitivity of the methods to density/connectivity variations modeled through speed.

Fig. 8a, depicts the relationship between vehicles average speed on the x-axis and the corresponding packet delivery rate on the y-axis. The data clearly shows an inverse relationship between the two variables, wherein PDR decreases gradually as vehicles speed increases. At the lowest speed of 10 km/h, PDR is at its highest level for all methods. However, there is a steady decline in PDR as speeds rise sequentially. This declining trend suggests that higher vehicle mobility negatively impacts network performance in terms of reduced packet delivery. The reduction in PDR with increasing speeds can be attributed to the fact that vehicles move away from each other faster and get out of range more frequently. This increased distance between vehicles at higher speeds leads to a higher probability of packet loss and transmission errors, resulting in a decrease in PDR. With the increase in vehicle speeds, the Proposed Method performs somewhat better than the greedy approach, which can be attributed to the optimal allocation of resources. As observed in Fig. 8a, the SLA-based method exhibits lower values when vehicle mobility and speed increase.

Fig. 8b, evaluates the SLA adherence rate based on the average speed of vehicles. Based on the chart, at a average speed of 10.00, the Greedy Algorithm and the Proposed Method have nearly similar results. As the average speed of vehicles increases, both methods show a gradual decrease in performance. However, the Proposed Method consistently outperforms the Greedy Algorithm at each level of average speed. It indicates that the Proposed Method is more effective in allocating resources and providing services compared to the Greedy Algorithm. Similar to the findings in Figure 8a, the SLA-based method exhibits a similar behavior and lower values as the speed increases.

The results obtained from varying the average speed provide valuable insights into the extent to which mobility affects performance of the evaluated methods. Comparing the slopes of the PDR and SLA adherence curves with increasing speeds reveals that vehicle mobility affects SLA adherence slightly more than PDR. This suggests that higher speeds could introduce larger delays in service delivery, impacting SLA to a greater extent. Additionally, a wider gap is observed between the Proposed Method and the others in terms of SLA adherence compared to PDR. This gap widens with higher speeds, indicating that the Proposed Method is better able to optimize service allocation under varying mobility conditions by leveraging location information in clustering vehicles and provider selection.

4) VARYING REQUESTERS-TO-PROVIDERS RATIO (R/P)

The Requesters-to-Providers ratio represents the ratio of requesters to providers in the network. For instance, if the number of requesters is half of the number of providers, the value would be 0.5. Conversely, if the number of requesters and providers are equal, the ratio would be 1. The goal of varying the ratio of requesters to providers is to evaluate the performance of the approaches under different event situations, ranging from when there are sufficient providers to support requests, to scenarios where resources are scarce and require careful planning to optimize provisioning.

In Fig. 9a, as the ratio of R/P increases, the PDR for all three approaches generally decreases. This is expected because with fewer requesters compared to providers, there is more resources to be selected, leading to higher PDR.

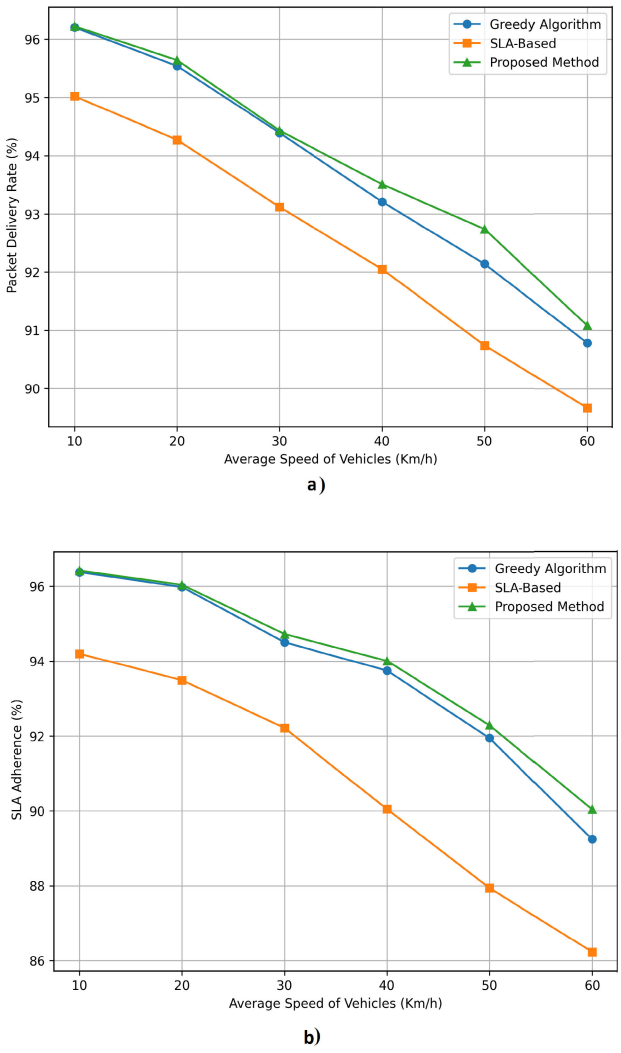

FIGURE 8. Performance comparison when varying the average speed of vehicles in terms of: a) PDR, b) SLA Adherence.

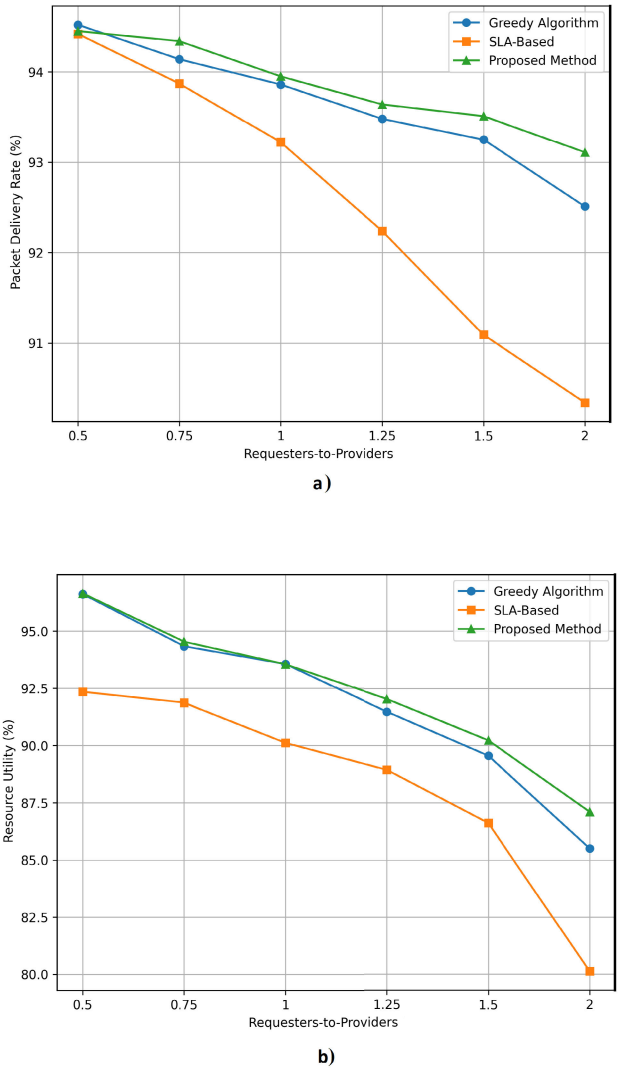

FIGURE 9. Performance comparison when varying the Requesters-to-providers ratio in terms of: a) PDR, b) Resource Utility.

However, the Proposed Method consistently outperforms the Greedy Algorithm and SLA-Based method across almost all R/P ratios in terms of PDR. The performance gap between the SLA-Based method and the other two approaches (Proposed Method and Greedy Algorithm) widens as the R/P ratio increases. This suggests that when resources are scarce (higher R/P ratio), the selection of proper providers becomes more crucial, and the SLA-Based method's selection criteria may not be as effective as the other two approaches. When the R/P ratio is low (e.g., 0.5), the performance of all three approaches is relatively close, indicating that when resources are abundant, the selection criteria may not significantly impact the overall PDR. In Fig. 9b, similar to the PDR trend, the Resource Utility for all three approaches decreases as the R/P ratio increases, reflecting the increased contention for resources when there are more requesters compared to providers. The Proposed Method consistently outperforms the Greedy Algorithm and SLA-Based method across all R/P ratios in terms of Resource Utility. This demonstrates its effectiveness in optimizing resource utilization while also maintaining high service provisioning performance, even under resource-constrained scenarios.

5) VARYING PARAMETERS ACROSS DIFFERENT SCENARIOS

In this section, we aim to evaluate the approaches under more dynamic scenarios to analyze their ability to handle changing network conditions. As the formulation contains coefficients, we also vary their values to analyze the impact on the Proposed Method, especially in specialized scenarios. It is important to note that since the sum of all coefficients must equal 1, in scenarios where a particular coefficient value varies, the values of the other coefficients will be adjusted accordingly and equally to maintain this constraint.

Moreover, in each scenario where we vary one parameter, the values of other parameters are randomly generated. For

example, when evaluating the impact of different delay tolerance levels (Delay Sensitive Service Requests rate), the service volumes, vehicle mobility patterns and other attributes will remain random. This allows isolating the effect of the varied parameter, while retaining randomness in other factors to better emulate real-world conditions.

a: MOBILITY IMPACT

In a realistic urban scenario, some vehicles will be parked while others are moving in different directions on the roads. In this subsection, we aim to evaluate the impact of such dynamic mobility patterns on the performance of the proposed approaches.

In Fig. 10a, as movement proportion rises, PDR declines sequentially. This downward trend establishes an inverse relationship between increasing vehicles movement and decreasing PDR. The progressive fall in network performance can be attributed to greater disruption of wireless connections and rising packet collisions or drops resulting from heightened vehicular activity. These findings provide valuable insights into how varying degrees of vehicular mobility influence the reliability of packet transmissions in vehicular networks. However, the Proposed Method consistently outperforms the Greedy Algorithm and SLA-based method. With increasing vehicle mobility, the algorithm employed in the SLA-based method leans towards utilizing mobile clouds over fixed clouds, leading to higher packet loss.

Fig. 10b, represents the SLA adherence rate comparison between the Greedy Algorithm, SLA-based method and the Proposed Method. The adherence rate is measured for different levels of vehicle movement. As the vehicle movement increases, all methods experience a slight decrease in the SLA adherence rate. However, the Proposed Method consistently outperforms the Greedy Algorithm at each level, albeit with diminishing differences. The reason for this is that the service allocation to service providers in the proposed algorithm is slightly better than the Greedy Algorithm. Therefore, mobility will also have a negative impact on this allocation. The SLA-based method, in terms of SLA adherence, also exhibits lower values when mobility is increased. This behavior is due to the algorithm's inclination to utilize mobile clouds more frequently than fixed clouds as mobility increases.

Finally, in Fig. 10c, we evaluate the impact of varying the coefficient related to the mobility of providers (β) on the performance of the Proposed Method. As shown, when the average vehicle speed is less than 30 km/h, setting $\beta = 0.2$ results in better performance. At these lower speeds, other factors like delay sensitivity and service volume have a similar effect on the network. Therefore, increasing the weight of mobility (which decreases the weights of other parameters) leads to lower packet delivery rate. However, as the average speed increases from 30 to 60 km/h, prioritizing mobility through $\beta = 0.6$ yields better performance for these scenarios. At higher speeds, the impact of mobility variations becomes more prominent. But the difference compared to $\beta = 0.2$ is not significant, indicating that the Proposed Method adapts well under different mobility conditions controlled by β.

b: DELAY SENSITIVE SERVICE REQUESTS

In this subsection, we evaluate the effects of varying the rate of delay-sensitive service requests on the performance of the different approaches. Delay-sensitive requests impose stringent delay constraints that must be met through optimal resource allocation. Analyzing how approaches adapt to scenarios with differing rates of such requests provides insights into their resilience under dynamic demand conditions involving priority requests.

Fig. 11a illustrates the SLA Adherence performance of the Proposed Method and the SLA-Based method across varying percentages of delay-sensitive service request rates, ranging from 20% to 100%. The Greedy Algorithm is omitted from this analysis, as mentioned, due to its insignificant difference from the Proposed Method's performance, similar to other scenarios. As the percentage of delay-sensitive service requests increases, the SLA Adherence decreases for both the Proposed Method and the SLA-Based method. This is expected because a higher proportion of delay-sensitive requests imposes stricter timing constraints, making it more challenging to maintain high SLA Adherence levels. It is evident from the chart that the Proposed Method consistently outperforms the SLA-Based method across all delay-sensitive service request rates. The performance gap between the two approaches widens as the percentage of delay-sensitive requests increases, with the Proposed Method exhibiting a slower decline in SLA Adherence compared to the SLA-Based method. The superior performance of the Proposed Method can be attributed to its ability to consider delay-tolerant applications directly in the problem formulation.

In Fig. 11b, as the percentage of delay-sensitive service requests increases, the Service Delay decreases for both the SLA-Based method and the Proposed Method. This behavior is expected, as a higher proportion of delay-sensitive requests necessitates using more fog/edge resources than conventional cloud resources to meet the stringent timing requirements of these requests. It is evident from the chart that the Proposed Method consistently outperforms the SLA-Based method across all delay-sensitive service request rates in terms of Service Delay. The performance gap between the two approaches widens as the percentage of delay-sensitive requests increases, with the Proposed Method exhibiting a more significant reduction in Service Delay compared to the SLA-Based method. The superior performance of the Proposed Method in minimizing Service Delay can be attributed to its strict formula for not using conventional cloud resources for delay-sensitive requests, as mentioned in Table 5.

In Fig. 11c, three different values of δ are considered: 0.2, 0.4, and 0.6, representing varying degrees of emphasis placed on the delay parameter in the Proposed Method's

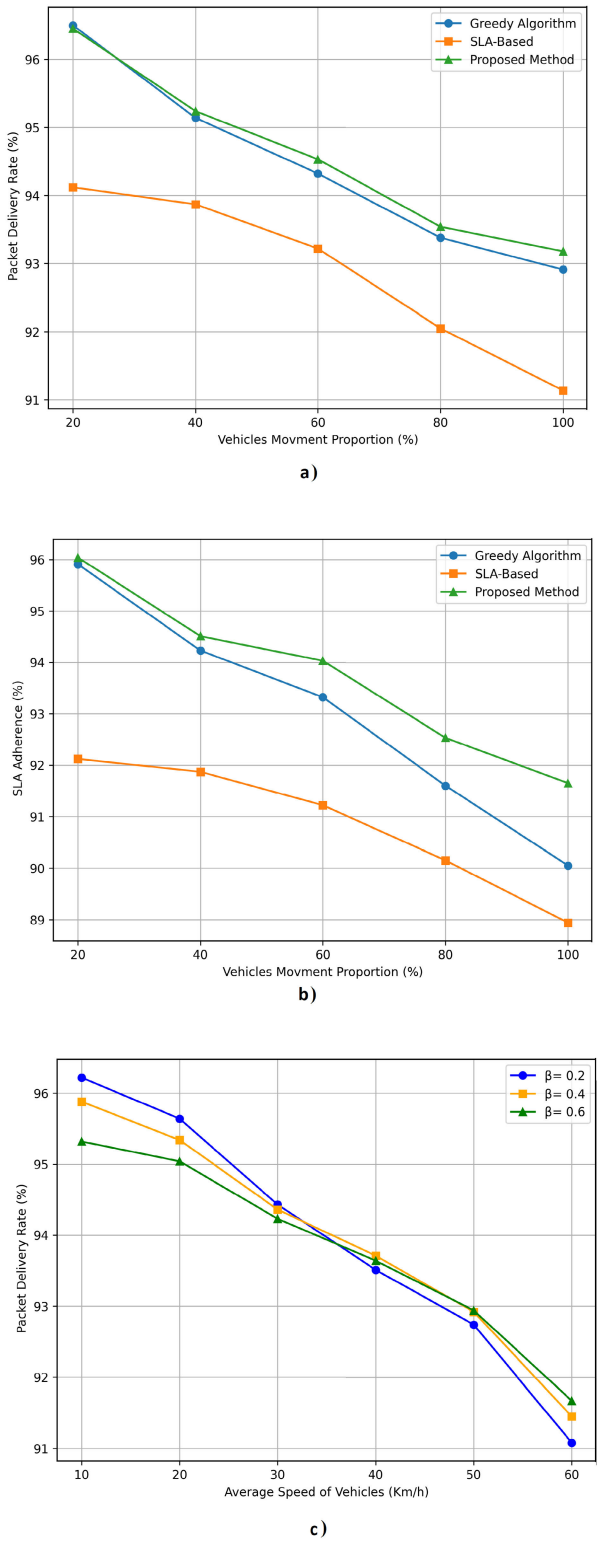

FIGURE 10. Performance comparison when varying the vehicles movement proportion in terms of: a) PDR, b) SLA Adherence, c) PDR with Different β.

formulation. As the percentage of delay-sensitive service requests increases, the SLA Adherence decreases for all three values of δ with the same reason for the Fig. 11a. When the percentage of delay-sensitive requests is less than 50% (e.g., 20% or 40%), the Proposed Method with $\delta = 0.2$ performs better in terms of SLA Adherence compared to the higher values of δ (0.4 and 0.6). This observation suggests that when the majority of requests are delay-tolerant, assigning a lower weight to the delay parameter ($\delta = 0.2$) leads to better SLA Adherence performance. Neglecting other important parameters by assigning a higher weight to the delay parameter can have a negative impact on SLA Adherence in such scenarios. However, as the percentage of delay-sensitive requests increases beyond 50% (e.g., 60%, 80%, or 100%), the trend reverses, and the Proposed Method with higher values of δ (0.4 and 0.6) starts to outperform the configuration with $\delta = 0.2$ in terms of SLA Adherence. When the majority of requests are delay-sensitive, prioritizing the delay parameter by assigning a higher weight ($\delta = 0.4$ or 0.6) becomes more important to meet the stringent timing requirements of these requests and maintain high SLA Adherence levels. In such scenarios, the impact of neglecting the delay parameter by assigning a lower weight ($\delta = 0.2$) becomes more pronounced, leading to a degradation in SLA Adherence performance. It is worth noting that while the Proposed Method with $\delta = 0.6$ performs slightly better than $\delta = 0.4$ for higher percentages of delay-sensitive requests (e.g., 80% and 100%), the difference in SLA Adherence performance between these two configurations is not significant.

Finally, as it is evident in Fig. 11d, similar to other charts, when the delay coefficient δ is set to 0.6, the Service Delay exhibits lower values compared to other δ values as the percentage of delay-sensitive service requests increases. This behavior is expected, as a higher value of δ places more emphasis on minimizing the delay parameter, leading to lower Service Delays, especially when the proportion of delay-sensitive requests is higher.

However, it is important to consider the trade-off between Service Delay and other parameters like SLA Adherence. While a higher value of δ (e.g., 0.6) may result in lower Service Delays, it could potentially compromise the overall SLA Adherence performance, as observed in the previous chart (Fig. 11c).

c: HIGH VOLUME SERVICE REQUESTS

Vehicular cloud networks support a wide range of applications, from safety-critical messages to resource-intensive services like live video streaming. These applications generate varying volumes of service requests, which can have a significant impact on the performance of the network and the resource allocation strategies employed. In this subsection, we evaluate the different approaches by varying the service volume, which represents the overall demand for resources and services within the vehicular cloud network. Moreover, based on the assumptions in the simulation, where every vehicle produces data ranging from 1 to 50 MB (in each period of 5 seconds), we can calculate the estimated data generated per vehicle per hour ranges from approximately 1 GB/h to 36 GB/h.

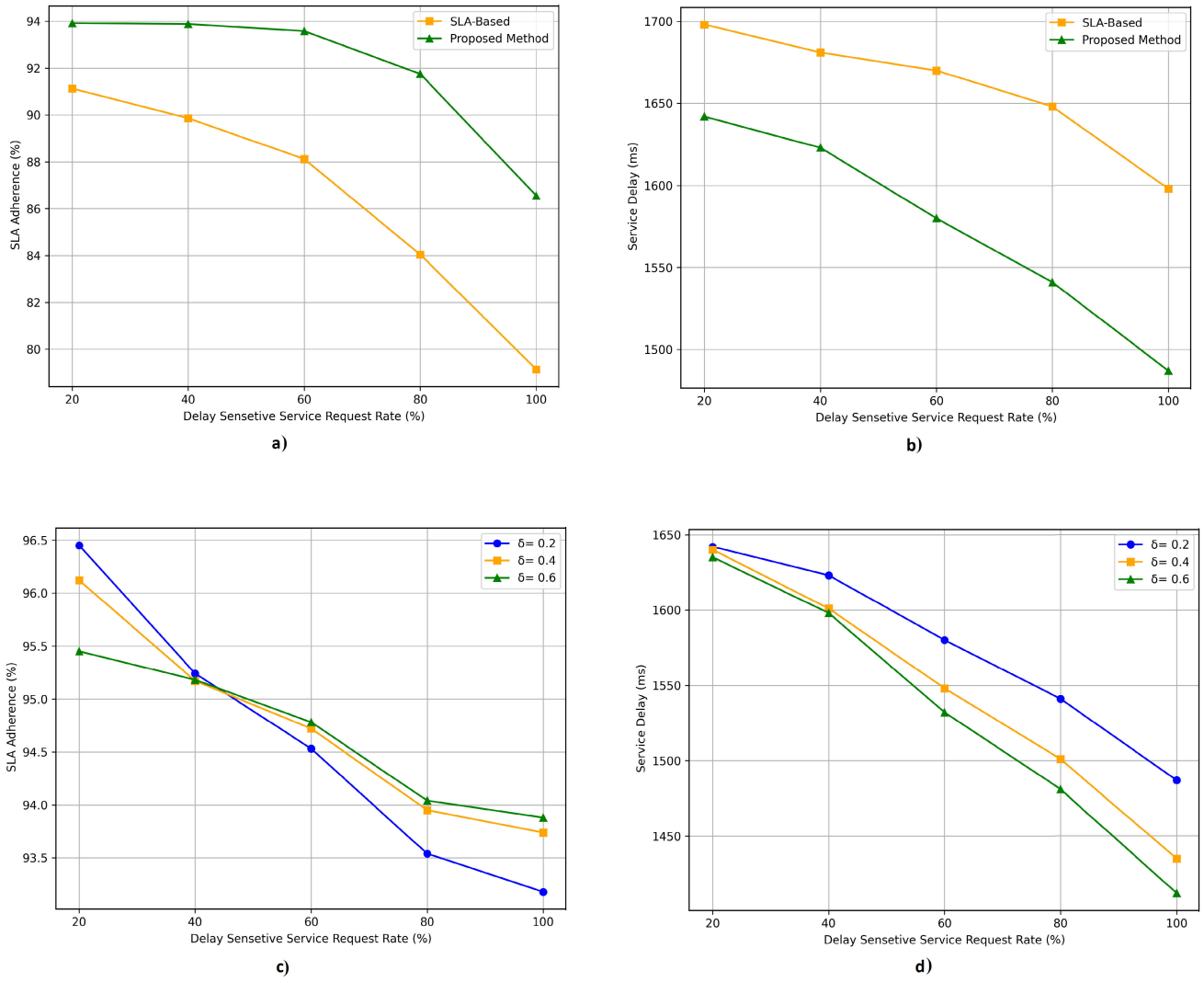

FIGURE 11. Performance comparison when varying delay sensitive service requests ratio in terms of: a) SLA Adherence, b) Service Delay c) SLA Adherence with different δ, d) Service Delay with different δ.

In Fig. 12a across all three approaches, a general trend is observed where the SLA Adherence decreases as the Average Service Volume increases. This behavior is expected because as the service volume grows, the network becomes more congested, and the likelihood of packet loss or service disruption increases, making it more challenging to maintain high levels of SLA Adherence. The Proposed method consistently outperforms the Greedy Algorithm and the SLA-Based approach across all service volumes in terms of SLA Adherence. This superior performance can be attributed to the Proposed Method's ability to consider factors such as the location and capacity of service providers when making resource allocation decisions.

Fig. 12.b evaluates the impact of varying the coefficient α related to service volume and provider capacity on the SLA Adherence performance of the Proposed Method. Three different values of α are considered: 0.2, 0.4, and 0.6, representing varying degrees of emphasis placed on service volume and provider capacity in the Proposed Method's resource allocation strategy. Initially, when the Average Service Volume is low (1 MB), the Proposed Method with $\alpha = 0.2$ exhibits the highest SLA Adherence compared to the higher values of α (0.4 and 0.6). This behavior is expected because at lower service volumes, other parameters such as delay become more crucial in maintaining high SLA Adherence levels. Assigning a higher weight to service volume and provider capacity by increasing α may lead to suboptimal performance when the service demands are relatively low. As the Average Service Volume increases from 1 MB to 10 MB and 20 MB, the trend reverses, and the Proposed Method with higher values of α (0.4 and 0.6) starts to outperform the configuration with $\alpha = 0.2$ in terms of SLA Adherence. This is because as the service volume increases, considering the volume and capacity parameters becomes

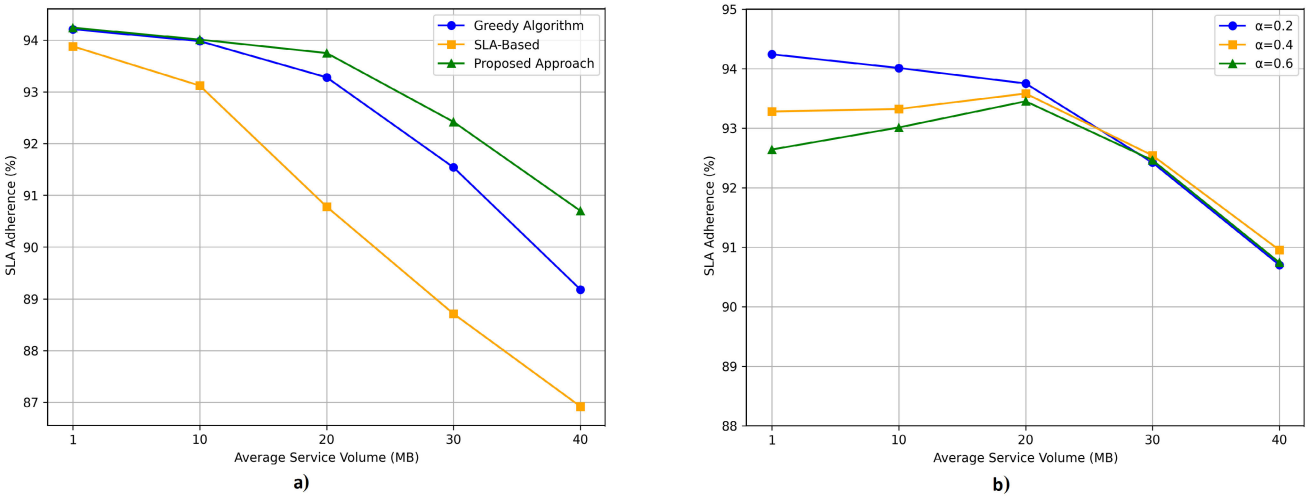

FIGURE 12. Performance comparison when varying the average service volume in terms of: a) SLA Adherence, b) SLA Adherence with different α.

more important to ensure efficient resource allocation and prevent network congestion, which can degrade SLA Adherence. However, at the middle range of Average Service Volume (around 20 MB), the SLA Adherence performance across all three values of α converges, with minimal differences observed. This convergence suggests that within this range of service volumes, the impact of varying α on SLA Adherence is relatively small, and other factors may play a more significant role in determining the overall performance. As the Average Service Volume continues to increase beyond 20 MB (e.g., 30 MB and 40 MB), a distinct trend emerges where the Proposed Method with $\alpha = 0.6$ exhibits the highest SLA Adherence, followed by $\alpha = 0.4$ and then $\alpha = 0.2$. This behavior can be attributed to the fact that at very high service volumes, prioritizing service volume and provider capacity by assigning a higher weight ($\alpha = 0.6$) becomes crucial to mitigate the impact of network congestion and resource constraints on SLA Adherence. The reason behind the descending trend in SLA Adherence for all values of α as the Average Service Volume increases can be attributed to the increased network congestion and potential packet loss associated with higher service volumes. As more service requests are processed, the risk of network saturation and service disruptions increases, leading to a decline in SLA Adherence, regardless of the value of α.

6) CONSUMER UTILITY

The evaluation of consumer utility in this part aims to assess the effectiveness of the different approaches in selecting appropriate pricing options for consumers while maintaining desirable levels of SLA factors. The primary objective is to demonstrate how well these approaches can balance the trade-off between providing cost-effective solutions for consumers and ensuring adherence to SLA requirements.

Fig. 13 presents the distribution of consumer utility values for each approach, providing insights into the central

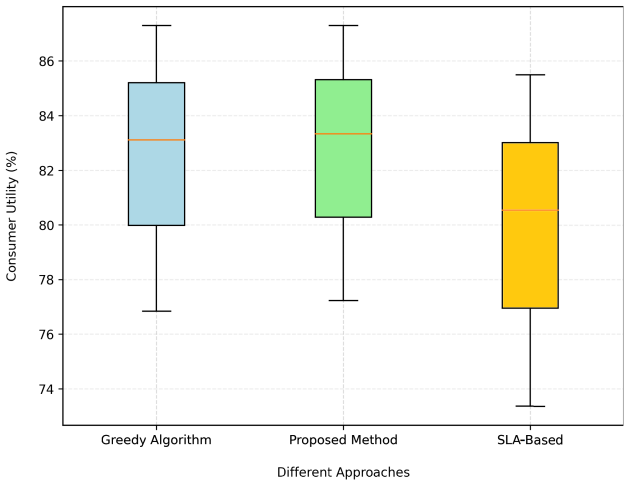

FIGURE 13. Performance comparison in term of consumer utility.

tendency (median), spread, and potential outliers. The use of a boxplot is particularly useful in this analysis because it captures the variability and potential outliers in consumer utility values. Since the scenarios generate random events and parameters (such as prices), the consumer utility values may vary more significantly. By depicting the minimum, maximum, and quartile values, the boxplot provides a comprehensive overview of the distribution of consumer utility for each approach, allowing for a more informed comparison. Based on the chart, the Proposed Method exhibits the highest median consumer utility among the three approaches, indicating that it generally performs better in terms of providing cost-effective solutions while maintaining desirable levels of SLA factors. The SLA-Based approach, on the other hand, has the lowest median consumer utility among the three approaches. Additionally, its boxplot has the largest spread between the minimum and maximum

values, suggesting a higher degree of variability and a wider range of consumer utility outcomes. This variability could be attributed to the specific method used by the SLA-Based approach for selecting pricing options, which may result in a broader range of consumer utility values depending on the scenarios encountered.

V. CONCLUSION

This paper presented a comprehensive service provisioning approach for vehicular cloud networks using fuzzy logic optimization and centralized orchestration. A mathematical model was formulated incorporating crucial factors like mobility, delay, cost, data volume and location suitability. Fuzzy logic techniques were leveraged to handle uncertainty in assessing provider-request compatibility. A heuristic algorithm was tailored to efficiently solve the NP-hard optimization problem. Simulations under diverse VCN scenarios evaluated the approach's effectiveness in maximizing the suitability of service provisioning. Comparisons with optimal solutions validated the quality of the proposed heuristic. Additional experiments demonstrated clear performance improvements over Greedy Algorithm and SLA-based method in key metrics like provisioning score, packet delivery rate, SLA adherence and resource utility. The important point to consider is that the PDR results for both proposed and Greedy Algorithm exhibit a similar and very close trend. However, this issue becomes more pronounced in the SLA adherence rate metric, especially with the increase in vehicle mobility and speed, where the Greedy Algorithm shows a greater reduction. This is because the SLA adherence rate metric is more stringent in its requirements.

While optimization methods such as simplex can significantly enhance service provision, especially in complex scenarios, it is crucial to consider the practical limitations faced in real-world applications. The processing constraints of vehicular boards, particularly when dealing with a high volume of vehicles, can lead to extensive computation times, potentially resulting in service delays. Balancing between the need for optimization and the practical constraints of real-world implementation is essential for achieving efficient and timely service delivery.

One of the primary assumptions made in this paper is the existence of RSUs. As part of our future work, we plan to expand our model to accommodate situations where RSUs are not present. This extension will concentrate on vehicular cloud networks that rely solely on vehicles for service provisioning. Additionally, incorporating new parameters, such as energy consumption, will be a crucial step in further enhancing our research. Morover, we will explore the utilization of machine learning algorithms to optimize parameters, as it presents a key area of focus for our research.

REFERENCES

[1] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.

[2] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *Int. J. Pervasive Comput. Commun.*, vol. 7, no. 1, pp. 7–21, Apr. 2011.

[3] S. K. U. Zaman, A. I. Jehangiri, T. Maqsood, Z. Ahmad, A. I. Umar, J. Shuja, E. Alanazi, and W. Alasmary, "Mobility-aware computational offloading in mobile edge networks: A survey," *Cluster Comput.*, vol. 24, no. 4, pp. 2735–2756, Dec. 2021.

[4] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 128–135, Aug. 2016.

[5] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, Feb. 2020.

[6] C. Li, X. Zuo, and A. S. Mohammed, "A new fuzzy-based method for energy-aware resource allocation in vehicular cloud computing using a nature-inspired algorithm," *Sustain. Comput., Informat. Syst.*, vol. 36, Dec. 2022, Art. no. 100806.

[7] A. H. Salem, I. W. Damaj, and H. T. Mouftah, "Vehicle as a computational resource: Optimizing quality of experience for connected vehicles in a smart city," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100432.

[8] S. K. Pande, S. K. Panda, and S. Das, "Dynamic service migration and resource management for vehicular clouds," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1227–1247, Jan. 2021.

[9] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25536–25545, Dec. 2022.

[10] G. Tang, D. Guo, K. Wu, F. Liu, and Y. Qin, "QoS guaranteed edge cloud resource provisioning for vehicle fleets," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5889–5900, Jun. 2020.

[11] C. Tang, S. Xia, Q. Li, W. Chen, and W. Fang, "Resource pooling in vehicular fog computing," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–14, Dec. 2021.

[12] F. Jafari Kaleibar and M. Abbaspour, "TOPVISOR: Two-level controller-based approach for service advertisement and discovery in vehicular cloud network," *Int. J. Commun. Syst.*, vol. 33, no. 3, Feb. 2020, Art. no. e4197.

[13] F. Jafari Kaleibar and M. Abbaspour, "SLA-based service provisioning approach in vehicular cloud network," *Cluster Comput.*, vol. 24, no. 4, pp. 3693–3708, Dec. 2021.

[14] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.

[15] K. Zrar Ghafoor, K. Abu Bakar, M. van Eenennaam, R. H. Khokhar, and A. J. Gonzalez, "A fuzzy logic approach to beaconing for vehicular ad hoc networks," *Telecommun. Syst.*, vol. 52, no. 1, pp. 139–149, Jan. 2013.

[16] B. Jamil, H. Ijaz, M. Shojafar, and K. Munir, "IRATS: A DRL-based intelligent priority and deadline-aware online resource allocation and task scheduling algorithm in a vehicular fog network," *Ad Hoc Netw.*, vol. 141, Mar. 2023, Art. no. 103090.

[17] A. Chebaane, A. Khelil, and N. Suri, "TimeCritical fog computing for vehicular networks," in *Fog Computing: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2020, ch. 17, pp. 431–458.

[18] Y. Wang, S. Liu, X. Wu, and W. Shi, "CAVBench: A benchmark suite for connected and autonomous vehicles," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 30–42.

[19] B. St. Amour and A. Jaekel, "Data rate selection strategies for periodic transmission of safety messages in VANET," *Electronics*, vol. 12, no. 18, p. 3790, Sep. 2023.

[20] P. Bezerra, A. Melo, A. Douglas, H. Santos, D. Rosário, and E. Cerqueira, "A collaborative routing protocol for video streaming with fog computing in vehicular ad hoc networks," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 3, Mar. 2019, Art. no. 155014771983283.

[21] J. Patil and N. Sidnal, "Comparative study of intelligent computing technologies in VANET for delay sensitive applications," *Global Transition Proc.*, vol. 2, no. 1, pp. 42–46, Jun. 2021.

[22] Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Trans. Comput.*, vol. c-26, no. 12, pp. 1182–1191, Dec. 1977.

[23] S. McCanne, S. Floyd. (1997). *Network Simulator NS-2*. [Online]. Available: http://www.isi.edu/nsnam/ns/

FARHOUD JAFARI KALEIBAR was born in Kaleibar, Iran, in 1990. He received the M.Sc. degree in information technology (enterprise architecture) and the Ph.D. degree in computer software engineering from Shahid Beheshti University, Iran, in 2014 and 2022, respectively. He has also worked in research-based industrial companies for three years, focusing on related topics. He is currently a Postdoctoral Fellow with Carleton University. His research interests include vehicular cloud networks, the Internet of Things, and software defined networking.

MARC ST-HILAIRE (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Polytechnique Montréal, in 2006. He is currently a Professor with the School of Information Technology with a cross-appointment with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. He is conducting research on various aspects of wired and wireless communication systems. With more than 150 publications, his work has been published in several journals and international conferences. His research interests include network planning and design, network architecture, mobile computing, and cloud computing. In addition to serving as a member of technical program committees of various conferences, he is equally involved in the organization of several national and international conferences and workshops. Over the years, he has received several awards, including the Carleton Faculty Graduate Mentoring Award, the Carleton Teaching Achievement Award, and several best paper awards. Finally, he is actively involved in the research community.