HARMONIC: Shapley Values in Market Games for Resource Allocation in Vehicular Clouds

DOI:
10.1016/j.adhoc.2023.103224

Document Version
Accepted author manuscript

Citation for published version (APA):
Ribeiro Jr., A., B. D. da Costa, J., P. Rocha Filho, G., A. Villas, L., L. Guidoni, D., Sampaio, S., & I. Meneguette, R. (2023). HARMONIC: Shapley Values in Market Games for Resource Allocation in Vehicular Clouds. *Ad Hoc Networks*, *149*, Article 103224. https://doi.org/10.1016/j.adhoc.2023.103224

Published in:
Ad Hoc Networks

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Ad Hoc Networks 00 (2023) 1–21
Ad Hoc Networks

HARMONIC: Shapley Values in Market Games for Resource Allocation in Vehicular Clouds

Aguimar Ribeiro Jr.[a,**], Joahannes B. D. da Costa[e], Geraldo P. Rocha Filho[b], Leandro A. Villas[e], Daniel L. Guidoni[c], Sandra Sampaio[d], Rodolfo I. Meneguette[a]

[a] *University of São Paulo (ICMC/USP) - São Carlos, Brazil*
[b] *State University of Southwest Bahia (DCET/UESB) - Vitoria da Conquista, Brazil*
[c] *Federal University of Ouro Preto (DECOM/UFOP) - Ouro Preto, Brazil*
[d] *University of Manchester (Dept. of Computer Science) - Manchester, UK*
[e] *University of Campinas (Institute of Computing) - Campinas, Brazil*

Abstract

Real-time allocation of resources to fulfill service requests from road vehicles is becoming increasingly complex, for two main reasons: the continuous increase in the number of Internet-connected vehicles on roads all over the world, and the emergence of complex and resource-greedy applications that require fast execution, often under limited availability of computational resources. While many resource allocation solutions to this problem have been proposed recently, these solutions rely on unrealistic scenarios and constraints that limit their practical use.

This paper presents HARMONIC, a Game Theory-based coalition game that aims to maximize resource utilization and dynamically balance resource usage across multiple Vehicular Clouds (VCs). HARMONIC employs a *Shapley* value-based strategy to determine the order of task allocation to available resources. It is built upon our proposed *Market Game* model, specifically designed to address resource allocation challenges in dynamic VCs. We conduct a comparative analysis with existing literature solutions under various scenarios and resource constraints to evaluate HARMONIC's performance. Our simulation results demonstrate that HARMONIC achieves resource allocation in fewer rounds and with fewer failures. Furthermore, it effectively distributes tasks to more VCs, improving load balancing and overall system efficiency.

Keywords: VANET, Vehicular Clouds, Resource Allocation, Load-balancing, Game Theory, Shapley Value

1. Introduction

The advancement of technology in the Automotive Sector is well perceived in recent improvements in the safety and experience of passengers traveling in road vehicles [1, 2]. These advances have brought new challenges to those working in this sector and a significant paradigm shift, particularly palpable to users of the so-called smart vehicles, of which self-driving and electric cars [3, 4] are examples. The growth in the number of smart vehicles has been such that, by 2025, there will be about 2 billion Internet-connected road vehicles around the world [5]. To cope with the pace of this growth, the automobile industry, in partnership with various technology industries, has been heavily

[*] Corresponding author

[**] Principal corresponding author

Email addresses: aguimarjr@usp.br (Aguimar Ribeiro Jr.), joahannes.costa@ic.unicamp.br (Joahannes B. D. da Costa), geraldo.rocha@uesb.edu.br (Geraldo P. Rocha Filho), leandro@ic.unicamp.br (Leandro A. Villas), guidoni@ufop.edu.br (Daniel L. Guidoni), s.sampaio@manchester.ac.uk (Sandra Sampaio), meneguette@icmc.usp.br (Rodolfo I. Meneguette)

investing in new technology to better resource vehicles, including sensors and chipsets, for greater processing power and fast communication [6, 7].

Through sensing and communication devices, vehicle connectivity is made possible. This allows the provision of road services via resource sharing and information exchange for addressing challenges related to urban mobility, as discussed in [1, 8]. Communication between vehicles and other devices is, thus, carried out through a Vehicular *Ad Hoc* Network, a.k.a. VANET [9], as it is referred to in the remainder of this paper. In VANETs, communication from Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) [10, 1], and Vehicle to Pedestrian (V2P), in some cases [6], is possible.

VANETs underpin Vehicular Cloud Computing (VCC) functionality to enable communication between vehicles and other connectable devices. This promotes a paradigm shift bringing Cloud computing capabilities to VANETs. In a Vehicular Cloud (VC), a group of vehicles is associated with a pool of shared resources, e.g., memory, processing power, and storage. This way, vehicles can offer their computing resources for sharing. In this environment, challenges associated with Quality of Service (QoS), on-demand service requests, and scalability gain relevance [11, 12, 4, 13]. Consequently, the need for resource allocation algorithms to manage the available and shareable resources is critical [14, 4, 15]. Furthermore, effective management of large resource pools and wide ranges of services in VCs is more likely to be achieved via exploitation of Edge Computing [16, 17]. The approaches associated with this paradigm are better suited for implementing aggregates of computational resources and faster resource allocation and acquisition for finer-grained adjustments of Cloud service requests [18, 19]. In the same path, an important concept in such environments is vehicular Cloud Controller (CC) [20], an entity on the network edge responsible for creating and maintaining VCs and managing resources available from vehicles.

An important VCs' challenge is providing reliable and off-the-shelf services and applications, especially for safety, traffic control, and entertainment applications in a highly vehicular environment [21, 22]. Thus, in VCs, fast and real-time decision-making in allocating resources to fulfill services is of paramount importance. Several resource allocation models in VCs have been proposed and are available in the literature. The vast majority of these models are based on a set of methods, being the following among the most widely adopted: greedy algorithms [23], meta-heuristics [24], combinatorial optimization [25], multi-objective optimization [26], dynamic programming and reinforcement learning [27]. Despite contributing various techniques, approaches, application contexts, and assumptions, this body of effort does not consider an important aspect of the resource allocation problem in VCs, namely, the nature of the objective pursued by each vehicle participating in the VC. This objective can be collaborative, that is, to cooperate with each other to maximize a common objective; or competitive, where each participant tries to maximize their own objective selfishly. It is worth pointing out that the objective of individual vehicles is particularly relevant in environments where trade-offs between multiple and conflicting service requirements need to be balanced, for example, throughput and delay, resource utilization, energy consumption, and security. Enforcing a balance between conflicting service requirements impacts on decision-making in resource allocation. In particular, it impacts on how data collected from multiple vehicles should be aggregated or combined to fulfill their objectives, which is scenario-dependent.

We argue that game theory [28] is a more suitable method for addressing the challenge mentioned above than the ones used in previous works. Game theory is fitted for analyzing and modeling scenarios where interactive decision-making is required, particularly in highly dynamic and unstable environments, such as VCs. It provides a rich body of mathematical models for optimizing complex problems in which multiple agents with contradictory goals compete for limited resources or cooperate to maximize their common interests [29].

To support our argument, we develop a resource allocation policy based on a mathematical method to maximize the number of fulfilled service requests in this work. In other words, we demonstrate we can infer *optimized* allocations of available resources to tasks in VCs by modeling the problem using game theory. A heuristic algorithm, HARMONIC (s*H*apley v*A*lues in ma*R*ket ga*M*es for res*O*urce allocatio*N* in veh*I*cular *C*louds), is implemented to organize VCs as a single *Market Game*, creating a coalition-based cooperative environment for maximizing available resource usage. Furthermore, HARMONIC considers different network service rate requests and communication ranges during simulations. In addition, it employs a general approximation method for computing Shapley Values, which is shown to be suitable for fast solutions, presenting an acceptable margin of error. We evaluate the solutions under varying service request rates, communication ranges, task cost/weight, availability of resources, and network load. Our simulation results show that HARMONIC achieves its goals by providing higher resource utilization with a better load average across VCs compared to widely used resource allocation algorithms found in the literature.

To summarize, the main contributions of this paper are described as follows: (i) proposal of a cooperative and dynamic resource allocation scheme across multiple vehicular clouds using a market game model; (ii) proposal of an approximate method based on Shapley Values for tackling the resource allocation problem in vehicular networks; and (iii) an extensive simulation of the proposed solution, varying parameters such as communication range, task request rate, task weight, and vehicle resources. The obtained results show that the proposed solution is able to fulfill a higher number of service requests as well as better distribute workload across a higher number of vehicular clouds than other state-of-the-art solutions while reducing not only the number of cycles required for task allocation but also service denial rates.

The rest of this paper is structured as follows. Section 2 reviews background material relevant to this paper. Section 3 presents recent related works. Section 4 formally defines HARMONIC's model. Section 5 describes the operating principle and details the target use case scenarios and the proposed algorithms. Section 6 details the methodology and discusses the obtained results. Finally, Section 7 summarizes the achievements and shows the future work directions.

2. Background

In this section, we formally define Coalitional Games, Markets, Market Games, and Shapley values. These concepts form the basis for the development of the mathematical model of HARMONIC.

2.1. Coalitional Games

A Coalitional Game (CG) is a typical cooperative game that aims to model situations where players may cooperate to achieve their goals [28]. It assumes players can form coalitions and engage in binding agreements that yield a certain profit. The only requirement is that players be capable of arriving at decisions and committing to those decisions. The worth of a coalition with transferable utilities is the maximal profit it can generate through the cooperation of its participants. Different disciplines extensively use CG theory and its theorems for modeling [30]. The definition of a CG is described as follows [28].

Definition 2.1 (Coalitional game)**.** A CG with transferable utility is a pair (N, v) such that:

- $N = 1, 2, ..., n$ is a finite set of players. A subset of N is called a coalition. The collection of all the coalitions is denoted by 2^N.
- $v : 2^N \to \mathbb{R}$ is a function associating every coalition S with a real number $v(S)$, satisfying $v(\emptyset) = 0$. This function is called the coalitional function of the game.

A coalition formed by all players is named a grand coalition. The real number $v(S)$ is called the worth of the coalition S. In such a game, the result produced by the coalition is independent of the actions of the players who are not members of S.

2.2. Market

The economic model we use in this work is a market with a set of producers $N = \{1, 2, \dots, n\}$ who l trade commodities that form a set denoted by $L = \{1, 2, ..., l\}$ [28]. Different types of goods can be produced, such as metals and food products. We assume that the quantity of each commodity is nonnegative, and a vector of commodities, called a bundle, is denoted by $x = (x_j)_{j=1}^{L} \in \mathbb{R}^L_+$. Each producer i has a bundle of commodities x_i, where $x_i, _j$ represents the quantity of commodity j in this bundle that the producer owns. They also have a "production technology" represented by a production function $ui : \mathbb{R}^L_+ \to \mathbb{R}$: if $x_i \in \mathbb{R}^L_+$. A producer i can produce the sum of money $u_i(x_i)$ and since the production technologies may differ from one producer to another, their production functions may also differ.

The authors [28] also assume that every producer i has an initial endowment that is a bundle $a_i \in \mathbb{R}^L_+$ of goods, and the producers can trade among themselves. So, the total bundle of goods available to the coalition is $a(S) := \sum_{i \in S} a_j \in \mathbb{R}^L_+$. The goal is to maximize the money they produce so that the coalition can allocate a bundle to each member a bundle $x_i \in \mathbb{R}^L_+$, subject to the constraint given by Equation 1. Finally, an allocation for a coalition S is a collection of bundles of commodities $(x_i)_{\in S}$ where $x_i \in \mathbb{R}^L_+$ for every producer $i \in N$, satisfying $x(S) = a(S)$.

$$x(S) = \sum_{i \in S} x_i = \sum_{i \in S} a_i = a(S) \tag{1}$$

Formally, a market is defined as follows:

Definition 2.2 (Market)**.** A market is given by a vector $(N, L, (a_i, u_i)_{i \in N})$ where:

- $N = \{1, 2, ..., n\}$ is the set of producers.
- $L = \{1, 2, ..., l\}$ is the set of commodities.
- $\forall i \in N$, $a_i \in \mathbb{R}^L_+$ is the initial endowment of producer i.
- $\forall i \in N$, $u_i : \mathbb{R}^L_+ \rightarrow \mathbb{R}$ is the production function of producer i.

2.3. Market Games

According to Maschler et al. [28], every market can be associated with a coalitional game, in which the set of players is the set of producers $N = \{1, 2, ..., n\}$, and the worth of each nonempty coalition $S \subseteq N$ is defined by the Equation 2. This way, the worth of coalition S is the maximal sum of money that its members can produce if they trade commodities among themselves, and the coalitional game $(N; v)$ is defined as the market game derived from the market $(N, L, (a_i, u_i)_{i \in N})$. In a vast number of coalitional game types, we have chosen market games to model the resource allocation problem because other works [31, 32] proved the existence of at least one Pareto-optimal solution that solves the model.

$$v(S) = \max \left\{ \sum_{i \in S} u_i \left(x_i\right) : x = (x_i)_{i \in S} \in X^S \right\} \tag{2}$$

2.4. Shapley Values

Shapley value is a single-valued solution concept for coalitional games [33]. Considering the grand coalition N will be formed, the Shapley value answers how the worth $v(N)$ will be divided among the coalition members. The notion of a marginal contribution, the amount by which the player's participation increases the worth of a coalition, is central to the definition of Shapley value. An axiomatic approach defines the Shapley value as the unique solution concept that satisfies the following properties [34]:

- **Symmetry**: interchangeable players i and j give the same marginal contribution to every coalition that does not contain them. Adding player i to a coalition is equivalent to adding player j to that coalition.
- **Null player**: if a player contributes nothing to any coalition, he is a null player and receives no part of the coalition's worth.
- **Additivity**: the worth of two independent games is the sum of the worth of each.
- **Efficiency**: when assumed that the coalition will form the grand coalition, its members entirely distribute the worth.

Let k be a player, and S be an arbitrary coalition that does not include player k, the Equation 3 gives the Shapley value of k, which is the sum of marginal contributions of k to every possible coalition formed by the permutation of the other players.

$$Sh_k(N; v) = \sum_{S \subseteq N \setminus \{k\}} \frac{|S|! \times (n - |S| - 1)!}{n!} (v(S \cup \{k\}) - v(S)) \tag{3}$$

Computing the exact Shapley values is generally $\mathcal{NP}$-hard, requiring approximation methods for non-trivial models [34]. Furthermore, it requires time $O(2^n)$ for n players, making it intractable for many players. So, in this work, we have employed a general approximation method [35] based on sampling that provides fast solutions with an acceptable margin of error. All simulations in this study have used a sample size of $n = 1000$.

3. Related Works

Several works have been proposed for efficient task allocation in VC environments. For example, Luo et al. [36] analyzed the processing delay and monetary cost for task allocation in VCs. In this work, a scheduling framework with communication and computation is established for VCs, considering tasks with different requirements. Thus, a multi-objective optimization problem is formulated to minimize delay and monetary cost. For this, a scheduling algorithm based on Particle Swarm Optimization (PSO) is proposed to obtain Pareto-optimal solutions. However, due to its bio-inspired approach, its convergence time can affect the solution's overall performance. Furthermore, the authors do not consider vehicular mobility a requirement in the scheduling process.

Hattab et al. [37] introduced a polynomial time complexity algorithm for task allocation in VCs with different computational resources. First, the algorithm classifies tasks according to completion and waiting time ratio. Afterward, it selects a subset of tasks with the lowest proportion and solves a sequence of Linear Programs. They formulate the bottleneck assignment problem, where the goal is to minimize the completion time of the allocated tasks in the available VCs. However, this work does not consider the mobility property of vehicles for VC formation, i.e., in their view, VCs are stationary, and the proposed algorithm considers only one VC.

Wu et al. investigated the problem of task scheduling and resource allocation optimization considering the effect of vehicular mobility in the VEC [38] environment. Specifically, the authors formulate the joint optimization problem in a Min-Max perspective to reduce the overall latency in scheduling tasks. Furthermore, considering a vehicle's relatively stable movement patterns in a short period, mobility prediction is considered for designing a mobility prediction-based scheme to obtain better results. However, the mobility model is unrealistic, as vehicles need constant acceleration during the task scheduling and resource allocation processes.

Pereira et al. [39] proposed the RELIABLE algorithm for resource allocation based on the Analytic Hierarchy Process (AHP). AHP is a multi-criteria mathematical method used to assist in making complex decisions. RELIABLE applies an influencing factor and subsequently uses a decision matrix to decide the best place for executing a service. The algorithm uses Mobile Edge Computing (MEC) features as parameters, calculates an influencing factor for each parameter, and then uses the decision matrix to select the best MEC. The authors applied RELIABLE in a road environment with processing-capable road infrastructure called Roadside Units (RSUs), each forming a MEC. Each MEC has three features, Mobility Prediction, Bandwidth, and Service Time. RELIABLE is executed when a vehicle needs Fog resources and, based on the number of resources of each MEC, decides if the service can be serviced and where to allocate the resources. However, the arrival of tasks in the system is based on the vehicle's time entering the RSU coverage area. Besides, vehicular mobility follows the Random Way Point model. That is, vehicular mobility and task arrival rate do not follow realistic models.

Da Costa et al. presented a task scheduling mechanism for VCs based on combinatorial optimization [40]. After a VC formation process, a controller located at a higher level in the network receives the requests for resources and schedules them for processing in the available VCs using a pseudo-polynomial algorithm for the 0/1 Knapsack Problem. However, the authors do not consider contextual aspects in the scheduling decision process. That is, the impact of vehicular mobility and the time requirements of tasks are disregarded.

Wei et al. [26] formulated resource allocation in the vehicular cloud as a multi-objective problem to jointly reduce blocking probability and cost. The authors proposed an improved version of the nondominated sorting genetic algorithm II (NSGA-II), named AC-INSGA-II. The AC-INSGA-II improves the initial population, mutation, and crossover probabilities to avoid local optimization and increase diversity. The focus of the proposed solution is to use the resources with load balancing efficiently. AC-INSGA-II was implemented considering a three-tier architecture with a Central Cloud, Road Side Units, and Vehicular Clouds. Therefore, the solution considers traditional cloud resources in data centers and those provided by vehicles in the vehicular network. The RSU receives a request and decides if it will be processed by the central or vehicular clouds considering the requirements. Also, it does not use edge computing, which can increase latency in service requests.

Lee et. al. [41] addressed the problem of resource allocation taking into account the vehicles' mobility, especially during peak hours. Due to hardware/software resource restrictions of vehicular fogs, a limited number of vehicles can use fog computing resources. The study considers slow-moving and parked vehicles as available resources and these vehicles could be used to process tasks with the goal of real-time response. Thus, the proposed solutions aim to minimize the latency of the task execution. In cases where scenarios and traffic patterns lack slow and parked vehicles, the proposed solutions use the available RSU to execute the task, and for computationally intensive tasks, the central cloud is also used. The authors proposed a Heuristic based on Reinforcement Learning (RL) for resource allocation. RL is used to understand the mobility pattern in order to better allocate the resource considering the vehicles' mobility. The proposed resource allocation mechanism is based on fog computing for vehicular networks, which uses RL and a heuristic algorithm to improve resource allocation efficiency. However, the model considers computational resources from remote clouds and does not use only the resources shared by vehicles in vehicular clouds.

Tang et al. [42] proposed a greedy strategy algorithm that considers a utility function for task allocation in VCs, called DbHA (Distance Based Heuristic Algorithm for Utility Optimization). DbHA considers parameters such as the computing and communication resources required by the tasks for the decision-making process and that these parameters are continuous variables. The authors calculate the Hessian matrix of a function with such variables and obtain the optimal values of computation and communication when the maximum value of the utility function is reached. After this process, the Euclidean distance between the optimum values and the values of each request is calculated and fed to a Heap to obtain the minimum distance efficiently. The task allocation mechanism is based on a genetic algorithm and Particle Swarm Optimization (PSO). The mechanism is compared to other approaches regarding network performance and resource utilization efficiency. In addition, the mechanism uses data center resources for resource allocation, which naturally can introduce an unwanted delay in the system.

Marques et al. [43] proposed a resource allocation policy based on game theory to maximize resource utilization in VCs by analyzing the Nash equilibrium of the proposed game. The authors formulate the problem as a non-cooperative game where players, in this case, vehicles, have strategies to offer or not a resource and consume or not a service. The main idea of the model is that both consumers and resource providers have some kind of gain. However, a Nash equilibrium is not necessarily a Pareto-optimal [29], which, depending on the chosen strategy profile, can lead to inefficient solutions. In addition, the work also does not consider variations in network request rates.

Yu et al. studied the sharing and management of bandwidth to mobile applications in vehicular cloud environments [44]. They proposed a coalitional game model to support the cooperation between service providers to share their idle resources. It considered different service request rates, and the results showed a greater computational resource utilization above 75% than in scenarios without cooperation. In order to create a cooperative resource allocation mechanism, vehicles communicate with each other and with base stations to share information about their resource needs. This communication aims to improve the allocation of network resources, such as bandwidth, memory, and processing capacity. However, this work also makes use of external resources, which is not feasible in some scenarios.

Ribeiro Jr. et al. [45] proposed the ORANGE resource allocation policy based on game theory for maximizing resource utilization in VCs. It considered varying service request rates and performed aggregation of different types of resources. By comparing ORANGE with other solutions found in the literature, the authors showed that ORANGE presented better results than the considered solutions regarding resource usage rates. However, since the evaluation was limited to resource usage ratings, their work ignored other impactful factors, including workload balancing across VCs and variations in transmission rates.

Da Costa et al. [25] proposed a mechanism that optimizes task allocation in vehicular clouds. MORFEU uses combinatorial optimization techniques to determine the best task allocation among vehicles, taking into account the vehicles' processing load, location, and connectivity with other vehicles. To evaluate its performance, the mechanism was compared with other existing methods in the literature, and the results showed that MORFEU can significantly reduce processing time and improve task allocation efficiency. However, this work does not consider different request rates or communication ranges.

As discussed in this section, although many related works propose methods for resource utilization, they have unrealistic assumptions and constraints that limit their practical use. Hence, in this work, we propose HARMONIC. This heuristic algorithm organizes VCs as a single market, thus creating a market game that aims to maximize the use of available resources, considering varying rates of service requests (SR) on the network and different communication ranges (CR) in dynamic scenarios. Furthermore, the proposed method uses Shapley Values to organize and distribute the worth of the market game among its participants and does not consider resources other than the vehicles'. HARMONIC is a Load-Aware Load Balancing (LALB) algorithm that considers from its inception that balancing resource utilization among the VCs is necessary and should respect the Shapley Values' properties presented in Section 2.4. Table 1 summarizes this section. The criteria used to compare and contrast the related work are described as follows: (i) *Vehicle resources only* indicates whether only the computational resources available from vehicles are considered in the task allocation algorithm, i.e., resources located in edge entities or cloud data centers are not taken into account; (ii) *Pre-aggregation* indicates whether resources shared by vehicles are aggregated prior to service requests or if they are aggregated only when requests arrive, i.e., when task allocation is performed; (iii) *Different SR* indicates whether varying service rates are considered in the evaluation of the proposed solution; (iv) *Different CR* indicates whether the proposed model uses different communication ranges when creating of vehicular clouds; this variation in the communication area directly impacts the pool of available resources to be used for resource allocation, as it will allow more or fewer vehicles to participate in a VC; and (v) *LALB* indicates whether the algorithm is aware of the distribution of resource utilization load among the VCs and not only the maximum load that each VC can handle.

Table 1: Summary of related works.

Work	Characteristics						
	Architecture	Vehicle' resources only	Pre-aggregation	Different SR	Different CR	LALB	Method
Luo et al. [36]	Decentralized		✓				PSO
Hattab et al. [37]	Centralized		✓				Queue
Pereira et al. [39]	Decentralized	✓	✓			✓	AHP
Da Costa et al. [40]	Centralized	✓					Optimization
Wu et al. [38]	Centralized		✓				Optimization
Wei et al. [26]	Decentralized		✓				Multi-Objective
Lee et al. [41]	Centralized		✓	✓			RL-Heuristic
Tang et al. [42]	Centralized		✓				Genetic algorithm
Marques et al. [43]	Centralized	✓	✓				Game theory
Yu et al. [44]	Centralized		✓	✓		✓	Game theory
Ribeiro Jr. et al. [45]	Centralized	✓		✓			Game theory
Da Costa et al. [25]	Centralized	✓					Optimization
HARMONIC	Centralized	✓		✓	✓	✓	Game theory

4. A Market Game Model for Resource Allocation in Vehicular Clouds

Given the definition of markets and how to convert them into market games in Section 2, this work uses analogies between commodities and computational resources; producers and VCs or CC, and coalitions and the set formed by the VCs and the CC. In addition, reallocating commodities in the market means in our model allocating tasks. The proposed model is derived from the market $(N, L, (a_i, u_i)_{i \in N})$ where:

- $N = \{1, 2, ..., n\}$ is the set formed by the VCs and the CC.
- $L = \{1, 2, ..., l\}$ is the set of weights of tasks T, where $l = \max\{weight(K)_{k \in T})\}$;
- $\forall i \in N$, $a_i \in \mathbb{R}^L_+$ is the vector of available slots for task allocation in VCs or the number of tasks ready to be allocated by the CC. In other words, a_{ij} represents the number of allocated tasks with weight j if i is a VC. Otherwise, if i is the CC, a_{ij} represents the number of tasks with weight j waiting to be allocated. The Equation 4 shows the initial values for the slots, where $|T_j|$ represents the number of tasks with weight j waiting for allocation by the CC.
- $\forall i \in N$, $u_i : \mathbb{R}^L_+ \to \mathbb{R}$ is the production function for the coalition members defined according to Equation 5, where $|T_{i,j}|$ represents the number of tasks with weight j allocated in the VC i.

In words, the worth produced by each VC i is equal to the sum of the total number of tasks allocated in this VC multiplied by their respective weights. Therefore, when the CC allocates tasks in the VCs, the total worth generated by the coalition can the equal to $\sum_{i \in N} u_i(x_i)$. Finally, in order to convert the proposed market to the model used in this work is required to meet Equation 2 subject to the computational resources Ω_i of each VC i. Formally the model is given by Equation 6.

$$a_{i,j} = \left\{ \begin{array}{ll} 0, & if\ VC \\ |T_j|, & if\ CC \end{array} \right\} \forall j \in [1, l] \tag{4}$$

$$u_i = \left\{ \begin{array}{ll} \sum_{j=1}^{l} j|T_{i,j}|, & if\ VC \\ 0, & if\ CC \end{array} \right\} \forall j \in [1, l], i \in N \tag{5}$$

$$\begin{array}{ll} \textit{Maximize} & \sum_{i \in S} u_i\,(x_i) \\ \text{subject to} & u_i\,(x_i) \leq \Omega_i \end{array} \tag{6}$$

5. HARMONIC

This section presents a system overview shared by all the different simulation scenarios in this work. In addition, it details the algorithms that compose HARMONIC and implements the model detailed in Section 4.

5.1. Target Scenarios

The target scenario considered in this work is described as follows. The network is composed by vehicles, roadside units (RSUs) and the vehicular Cloud Controller (CC). The RSUs are deployed all over the map covering the whole road network area and acts as a relay between vehicles and the CC. The vehicular Cloud Controller is deployed in the network edge jointly with a RSU or one RSU could act as the Cloud Controller. Due to the high mobility of the vehicles, CC executes a clustering algorithm responsible for grouping vehicles into Vehicular Clouds (VC). Then the available computational resources inside a VC can be shared and allocated by the CC to tasks waiting for resources to run. This work considers the deployment of one CC and the optimal deployment of RSUs and CC is not the focus of this work.

Every vehicle contains an On-Board Unit (OBU) that enables communications with other vehicles and RSUs. The RSUs are responsible for gathering the vehicles' information in real-time and forwarding the demand for resources to the CC. When a vehicle requires more resources than it can provide for running a set of tasks, it can request these extra resources from the CC. The CC runs the HARMONIC algorithm, which determines the best allocation scheme for the tasks in the available VCs. The algorithm aims to maximize resource utilization considering the capacity of each VC in the vehicular network.

5.2. Algorithms

As shown in Algorithms 1, 2, and 3, we designed three algorithms to complete the entire resource allocation process in the VCs. Algorithm 1 is the main algorithm responsible for defining which cloud has its resources allocated to each task. Algorithm 2 details how the resources are allocated for the tasks in the market game. Algorithm 3 is the coalitional function *vFunctionVC* that determines the worth of allocating all tasks in a given group of VCs. The Shapley Value approximation method mentioned in Section 2 uses those functions to get Shapley Values according to Equation 3.

Firstly, Algorithm 1 creates the coalition among the VCs and the CC (line 2) and defines the initial endowments for the coalition participants (line 3) based on Equations 4 and 5 of the proposed model. Next, it highlights CC's initial endowments (line 4) since they represent the resources required for the tasks waiting for allocation. Then it generates a matrix to store the history of all allocated bundles in each of the participants (line 5), initializes the worth of the coalition (line 6), and the list of assigned tasks (line 7). Then, it creates the list that stores the Shapley Values that will be computed for VCs (line 8) and the tasks list (line 9).

After that, the Shapley Values for VCs are calculated (line 10), the algorithm sorts VCs in descending order by their respective computed Shapley Values (line 11), and tasks are sorted (line 12) in descending order by their weights (number of resources needed by the tasks for their complete execution). These are the points where HARMONIC mainly differs and extends the work presented in ORANGE [45]. For example, ORANGE always uses the VC with more available resources as the first to be used for allocation. On the other hand, HARMONIC considers Shapley Values to determine in which order the VCs will be used based on their marginal contribution generating a greater worth by the coalition and consequently allocating more tasks. Then for every VC according to its Shapley Value (line 13) and while tasks are waiting for allocation (line 14), the exchange of resources in the market occurs (line 15). Then finally, VC's allocation is updated (line 15), as well as the contribution of the VC to the overall worth of the coalition (line 16). It also updates the remaining resources (line 17), the list of tasks waiting for allocation (line 18), and the list of allocated tasks (line 19). The algorithm returns the worth of the coalition (line 22).

In Algorithm 2, the resources required by the tasks waiting for allocation, a list of those tasks sorted by their weights, and the available resources of the chosen VC are provided as input to the *MarketSHAP* function. This

Algorithm 1: HARMONIC ALGORITHM

```
   Input: $T$ - set of tasks, $V$ - set of VCs
   Output: $U$ - The worth generated by tasks $T$
 1 begin
 2     $C \leftarrow V \cup \{CC\}$
 3     $A \leftarrow$ InitialEndowments($C, T$)
 4     $Ac \leftarrow A[N]$
 5     $X \leftarrow \vec{0}$
 6     $U \leftarrow 0$
 7     $S \leftarrow \emptyset$
 8     $SortedVCs \leftarrow \emptyset$
 9     $SortedTasks \leftarrow \emptyset$
10     $VcShap \leftarrow$ CalcApproxSV($T$, vFunctionVC)
11     $SortedVCs \leftarrow$ SortDesc($V, by = VcShap$)
12     $SortedTasks \leftarrow$ SortDesc($T, by =$ Weight)
13     for $VC$ in $SortedVCs$ do
14         if $sum(Ac) \neq 0$ then
15             $X[i], Ui, Au, Ta \leftarrow$ MarketSHAP($Ac$, $VC.resources$, $SortedTasks$)
16             $U \leftarrow U + Ui$
17             $Ac \leftarrow Ac - \{Au\}$
18             $T \leftarrow T - \{Ta\}$
19             $S \leftarrow S \cup \{Ta\}$
20         else
21             break
22     return $U$
```

function initializes a matrix to store the history of all allocated bundles in the VC (line 2). Then the worth yielded by VC (line 3), the list of given tasks (line 4), and the resources still required by the jobs in the queue for allocation (line 5). Following the given list of tasks sorted by weights in descending manner (line 6), the algorithm finds out the weight of the current job (line 7). Next, it checks if exists enough resources in the VC for allocating the task (line 8). Otherwise (line 10), the task is skipped, and the next is considered (line 9). Once there are sufficient resources, the task is allocated, and the worth generated is increased by the task weight value (line 11). Next, the list of assigned tasks is updated (line 12), so it is the history of utilized resources (line 13), and the map of required resources is decreased (line 14). Ultimately, the code returns the allocated tasks and the used resources in the VC, the worth generated by the task allocation, and the remaining required resources not allocated (line 15).

Finally, Algorithm 3 is the coalitional function used in Equation 3 for Shapley Values computing. The algorithm initializes the worth that will be calculated (line 1). Then, it iterates over each task (lines 2-10) and, at each iteration, determines which VCs have enough idle resources (lines 3-4). Every VC with enough resources is eligible to allocate the task. If the available resources in VCs are insufficient, the current job is skipped, and the next is considered (lines 5-6). Otherwise (line 7), the function randomly picks one of the VCs with enough resources (line 8). Then, decreasing the VC's idle resource counter (line 9), increasing the worth (line 10), and repeating the process until there are no tasks (line 2) or enough available resources (lines 5-6) and returns the computed worth (line 11).

Since the algorithms were designed to compute the Shapley Values for each VC and use them to define the ordering of consuming available VCs' resources, HARMONIC prioritizes the usage of VCs with more significant marginal contributions to the overall worth of the market while allocating tasks.

Algorithm 2: MarketSHAP ALGORITHM

```
Input: $Ac$ - resources needed, $\Omega$ - resources of the chosen VC, $SortedTasks$ - tasks to be allocated
Output: $X$ - allocated resources in VC, $Ui$ - The worth generated by tasks $T$, $Ta$ - set of allocated tasks
begin
    $X \leftarrow \emptyset$
    $Ui \leftarrow 0$
    $Ta \leftarrow \emptyset$
    $Au \leftarrow Ac$
    for $T$ in $SortedTasks$ do
        $W \leftarrow \texttt{Weight}(T)$
        if $W > \Omega$ then
            continue
        else
            $Ui \leftarrow Ui + W$
            $Ta \leftarrow Ta \cup \{T\}$
            $X[W] \leftarrow X[W] + 1$
            $Au[W] \leftarrow Au[W] - 1$
    return X, Ui, Au, Ta
```

Algorithm 3: vFunctionVC - VCs COALITIONAL FUNCTION

```
Input: $T$ - set of all tasks, $Vc$ - set of VCs in coalition
Output: The worth generated by tasks $T$
$worth \leftarrow 0$
for $task$ in $T$ do
    $weight \leftarrow \texttt{Weight}(task)$
    $availableVCs \leftarrow \texttt{Filter}(Vc, by = \texttt{IdleResources} \geq weight)$
    if $\texttt{Len}(availableVCs) = 0$ then
        continue
    else
        $chosenVC \leftarrow \texttt{Choice}(availableVCs)$
        $\texttt{DecIdleResource}(chosenVC, weight)$
        $worth \leftarrow worth + weight$
return $worth$
```

6. Performance Evaluation

This section presents the methodology to evaluate HARMONIC's model. It includes the simulation setup, metrics and analyzes target scenarios and results.

6.1. Simulation Setup

The experiments used the Simulation of Urban MObility (SUMO) [46], version 1.12.0, joined with TAPAS Cologne [47] Scenario. The "TAPAS Cologne" simulation scenario describes the traffic within the city of Cologne (Germany) for the entire day. It is assumed to be one of the most extensive freely available traffic simulation datasets. The algorithms were implemented in Python using the Traffic Control Interface (TraCI) library to connect to SUMO. We have decided to evaluate HARMONIC between 9 and 10 a.m, right after a peak hour, due to the lower vehicular density on the map, where fewer resources provide challenging environments. However, even considering a non-peak hour, the traffic on the map during the selected period requires a powerful processing server to perform the simulations. For this reason, we scaled the traffic at 0.3 factor and used SUMO in mesoscopic mode. Figure 1 shows the structure of demand data over the period.

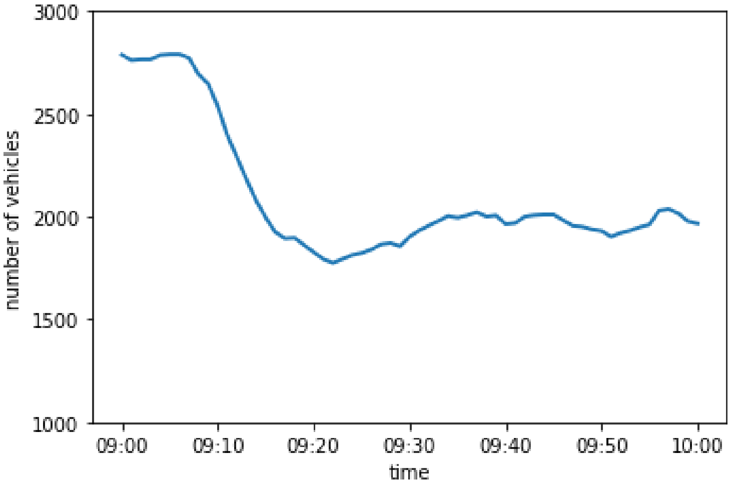


Figure 1: Number of vehicles over the selected interval (9-10 a.m.).

The mesoscopic mode refers to an intermediate level of mobility resolution between the microscopic and the macroscopic modes of urban mobility. The microscopic mode simulates the vehicle movements on the streets, and the macroscopic mode simulates the flow of vehicles. In the mesoscopic mode, vehicles are treated as discrete entities, and their interactions with other vehicles and the road infrastructure are modeled using simple rules based on traffic flow and driver behavior. In this case, simulations can be executed on a scale more significant than the microscopic mode while still capturing essential features of urban traffic. We use the mesoscopic mode to explore the high-level details regarding the considered mobility trace [47]. We also used the SUMO Intelligent Driver Model, which is a "car-following" model that describes the behavior of vehicles when following other vehicles on the road. This model takes into account the speed, distance and acceleration of vehicles to determine the appropriate acceleration and deceleration to safely and efficiently follow the vehicle ahead.

In a vehicular network, clustering is a process of grouping vehicles into vehicular clouds to share their resources and serve more requests on the network. The clustering interval is when these clouds are rearranged, and vehicles may be included or excluded from the vehicular cloud. In this way, to perform the clustering process, it is necessary to use a clustering algorithm, which can be based on different techniques, such as distance or density. The communication range is essential in this process, as it defines the maximum distance between vehicles so they can be considered to belong to the same vehicular cloud. Besides, it is necessary to define the task distribution in the vehicular network, that is, how often resource requests arrive in the system. The service rate is also a relevant factor, as it determines the number of task requests arriving into the vehicular network in a given period. Each task has a weight representing the total number of resources needed for its allocation, and this weight is considered when defining which tasks will be allocated in which vehicular clouds. The simulation time is the total time defined by the TAPAS Cologne scenario.

The vehicular network scenario considered during the simulations includes the vehicles, the roadside units, and the vehicular Cloud Controller (CC). Vehicles move around the road map following the mobility trace and can communicate with roadside units. When vehicular clouds are created, the vehicles share their resources with the roadside units. Also, the roadside units transfer the set of tasks to be executed to the vehicles. The roadside units act as gateways between the vehicles and the cloud controller. The cloud controller knows the network's supply and demand for resources, and this entity is responsible for executing the resource allocation algorithm. Also, the cloud controller manages the vehicular cloud formation process and ensures that resources are appropriately allocated.

This work also assumes that since CC allocates the resources to the task, it has all information it needs to execute and complete allocation. Hence, its deadline (system runtime) is equal to 1. In addition, due to the high mobility in the scenario, CC runs DBSCAN [48] algorithm every 60 seconds for creating the VCs where each vehicle can share up to 5 resource units. Different communication ranges (in meters) are considered $\{10, 25, 50, 75, 100, 150, 200\}$, the tasks' arrival rate follows a Poisson distribution of different average request rates in $\{20, 50, 100, 150, 200, 250\}$, and the tasks' weight follows uniform distributions ranging from $[1, \mu]$, where $\mu = \{25, 50, 75, 100, 150, 200, 250\}$. In this study, we compared six algorithms across a total of 1470 test scenarios, with each scenario executed 33 times. This comprehensive analysis considered all the combinations of different parameters listed in Table 2, allowing us to capture a wide range of variations in the experimental setup. We employed the Friedman test [49] in conjunction with the Nemenyi test [50, 51, 52] to assess the performance differences among the algorithms. The Friedman test provided insights into whether there were significant variations in the algorithms' performance across the numerous test scenarios. Upon identifying significant differences in the Friedman test, we proceeded with the Nemenyi test to conduct pairwise comparisons between the algorithm pairs. By doing so, we could determine which algorithms exhibited statistically significant disparities in their performance. For the Friedman and Nemenyi tests, a significance level of $p < 0.05$ was chosen, indicating that differences with p-values below this threshold were deemed statistically significant.

Table 2: Simulation parameters

Parameter	**Value**
Communication range	10, 25, 50, 75, 100, 150, 200 meters
Task arrival mode	Poisson distribution
Request rate (λ)	20, 50, 100, 150, 200, 250
Task's weight	25, 50, 75, 100, 150, 200, 250
Simulation time	1h TAPAS Cologne Scenario (Interval: 9-10 a.m.)
Clustering algorithm	DBSCAN → eps = communication range, min_samples = 2
Clustering interval	60 seconds
Resources by vehicle	1, 2, 3, 4, 5

Table 3: Notation Definitions

Variable	**Definition**
VC	Set of all available VCs
UVC	Set of VCs which allocated at least one task
T	Set of all tasks
T^A	Set of allocated tasks
$weight(T)$	Number of resources needed by task T
$resource(VC)$	Available resources in VC

To evaluate the solution, we considered the notations in Table 3, denoting UVC as the set of VCs effectively used for allocating at least one task and AT as the percentage of allocated tasks in a scenario. They are defined as follows:

$$UVC = \bigcup_{i=0}^{|VC|} VC_i \times k \tag{7}$$

$$AT(\%) = \frac{\sum_{i=0}^{|T^A|} weight(T_i^A)}{\sum_{j=0}^{|T|} weight(T_j)} \cdot 100 \tag{8}$$

where $k = 1$ indicates that VC_i is used and $k = 0$ indicates that VC_i is not used.

Furthermore, to demonstrate the effectiveness of using Shapley Values for resource allocation, we have counted the number of rounds (Nr) required to allocate AT and the number of allocation failures (Nf) that occurred during this process. A round is considered to happen every time the algorithms attempt to allocate a task i (T_i) to a vehicular cloud j (VC_j). An allocation failure is considered to have occurred when this attempt is unsuccessful. Next, we evaluated the allocation rate by the number of rounds (TAr), defined by Equation 9, in different scenarios. The higher the value of TAr, the more efficient the resource allocation is, as it indicates a higher percentage of resources allocated per round. The reduced number of rounds required to allocate the same amount of resources can lead to energy savings and lower latency in the network. We also considered the allocation rate by the number of allocation failures (TAf), defined by Equation 10. A higher value of (TAf) indicates more efficient resource allocation with fewer allocation failures. This reflects the effectiveness of the allocation algorithm in successfully assigning resources to tasks, leading to improved system performance and reduced resource wastage.

$$TAr = \frac{AT}{Nr} \tag{9}$$

$$TAf = \frac{AT}{Nf} \tag{10}$$

Finally, we denoted NTL as the ratio between the total resources needed to meet all service requirements and the available resources in VCs. It is expressed in percentages and is defined as follows:

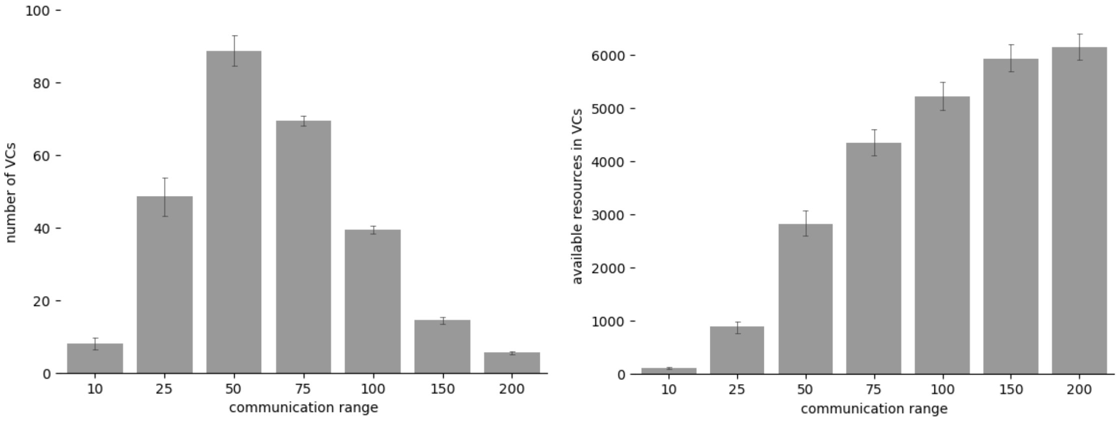
$$NTL = \frac{\sum_{i=0}^{|T|} weight(T_i)}{\sum_{j=0}^{|VC|} resource(VC_j)} \cdot 100 \tag{11}$$

We compared the proposed HARMONIC with five other mechanisms considering the task allocation rate, number of rounds, number of allocation failures, and load-balancing in a VC environment, namely *(i)* ORANGE [45] that uses game theory approach to allocate tasks in VCs as cited in Introduction 1; *(ii)* MORFEU [25] that uses a combinatorial optimization approach to allocate tasks; *(iii)* GREEDY and *(iv)* GREEDY-N, the last one based on Nabi *et al.* [53], and both use a greedy approach for task allocation; however, GREEDY-N considers more than one VC in the resource allocation process; and *(v)* DP, which employs a dynamic programming approach. MORFEU and GREEDY-N consider all existing VCs while defining the best task allocation scheme, and GREEDY and DP consider only the VC with more available resources. All those algorithms first utilize the VC with more resources to begin the allocation, while HARMONIC employs Shapley Values computing to choose which VC will be used first. For all performed analyses, when we evaluate one simulation parameter, we use all different values for the others. For instance, when the results show different values of communication range, for each communication range we compute the average for all values of request rate, task weight, and resources by vehicles. In this case, we can verify the impact of the communication range for all parameters and values and not for specific ones, which may not represent the entire scenario.

6.2. Simulation Results

In this section, we analyze the target scenarios and evaluate the performance of HARMONIC's resource allocation strategy compared to five other algorithms described in the literature.

Figure 2 shows the average number of VCs created and available resources in VCs under scenarios with different communication ranges. The clustering algorithm defines the VCs and their participating vehicles. Thus, all the task allocation algorithms share the resulting VCs and their available resources. In Figure 2a, it is worth noting that with the increase in communication range, the number of VCs created in the network also increases, reaching its peak when the communication range is equal to 50 and then starts to decrease. The number of formed VCs increases as the communication range increases because more cars are close to each other inside the given range. When the communication ranges vary from 75 to 200, the number of formed VCs decreases because the coverage is big enough to embrace many cars inside the same VC. Figure 2b shows the increasing quantity of available resources in VCs for different values of communication range. As explained in Figure 2a, the more the communication range increases, the more cars are clustered together. Consequently, more resources are grouped in VCs, and their resources are available for sharing.



(a) Number of VCs created

(b) Available resources in VCs

Figure 2: The average number of VCs created given by the clustering algorithm and available resources in VCs under scenarios with different communication ranges.

Figure 3 shows the average number of VCs engaged in resource allocation ($|UVC|$) under scenarios with different communication ranges. In all scenarios, HARMONIC is the method that engages a more significant number of VCs in task allocation. Because of the Shapley Values usage, HARMONIC fairly distributes the tasks by evaluating their weights among the coalition's members. In this case, the axioms, revisited in Section 2, define the concept of fairness.

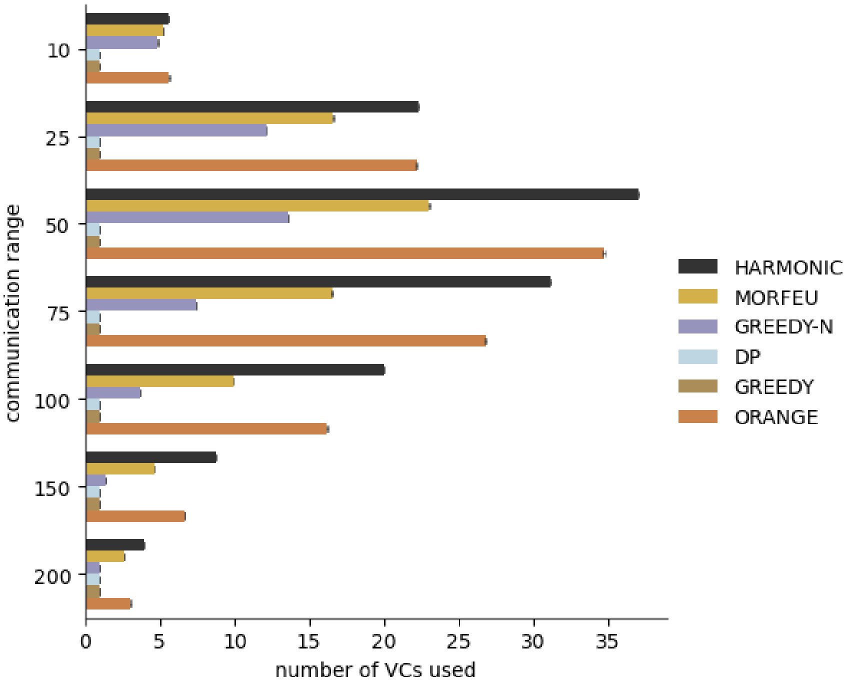
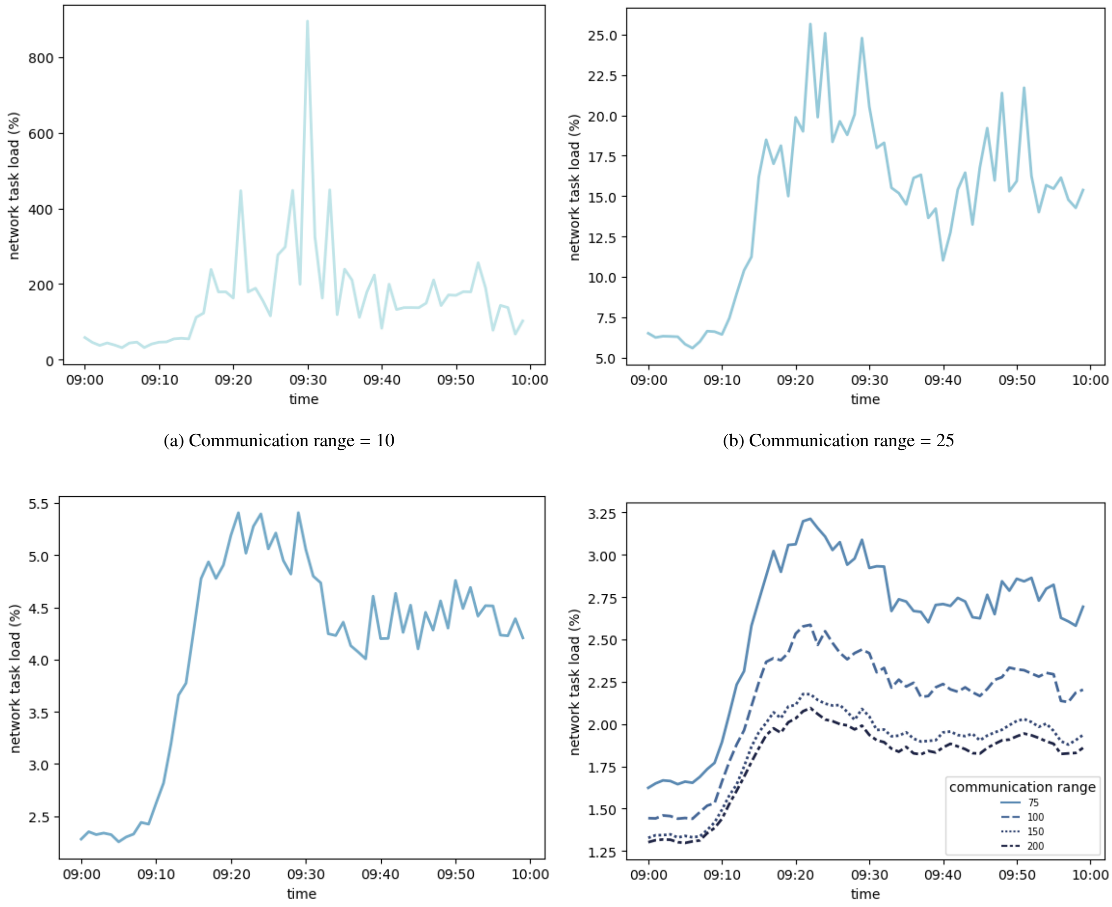


Figure 3: Average number of VCs used (UVC) under scenarios with different communication ranges.

Figure 4 shows the network task load (NTL) given by Equation 11 under scenarios with different communication ranges. As expected, with the increase in the communication range, the NTL decreases since there are more available resources in the network to meet the task load requirements. The demand for resources is more than 700 times ($NTL = 800\%$) higher than the available resources on the network (Figure 4a at 09:30). The value of NTL continues to decrease as coverage increases over time (Figures 4b, 4c, and 4d), but does not approach $NTL = 1$, which would indicate that the demand equals the supply of resources. Those figures suggest that the scenarios have severe resource constraints and are insufficient to meet all service requests.



(a) Communication range = 10

(b) Communication range = 25

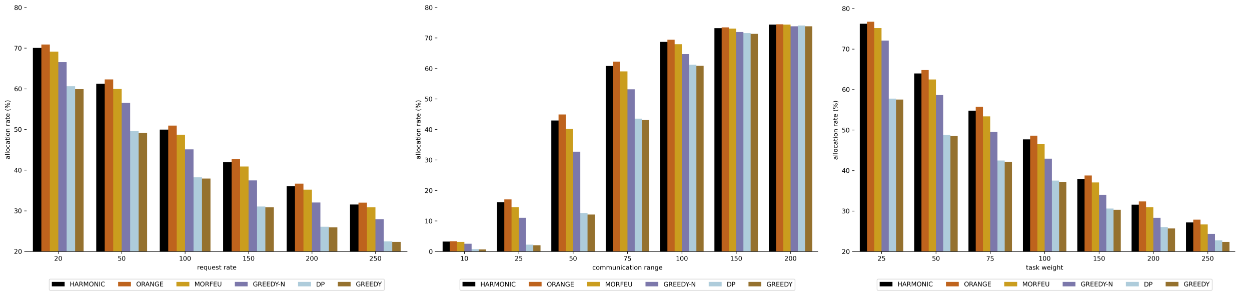
(c) Communication range = 50

(d) Communication range = [75,100,150,200]

Figure 4: Network task load (NTL) under scenarios with different communication ranges.

After conducting the Friedman and post hoc Nemenyi tests with a significance level of $p < 0.05$, we analyzed the algorithms to evaluate the allocation rate (AT), allocation rate per round (TAr), and allocation rate per failure (TAf). Among the 1470 evaluated scenarios, 410 scenarios exhibited significant performance differences among the algorithms. Table 4 presents the number of significant scenarios for each metric, with 332 scenarios (80.98%) indicating the dominance of the HARMONIC algorithm. When considering only the allocation rate (Figure 5), there is no significant difference in the dominance of one algorithm over the others. However, when examining the number of rounds (Figure 6) and allocation failures (Figure 7) during the allocation process, the differences between the HARMONIC algorithm and the others become evident. We explored the data from three perspectives for each metric: Increasing request rates, communication ranges, and task weights.

In Figure 5, we evaluated the allocation rate (AT) for the 272 scenarios in Table 4 (row 1). In Figure 5a, we observe the evolution of the metric by increasing the request rate. As the request rate increases, there is a decrease in the allocation percentage. We can attribute this decrease to the growing volume of tasks in the system and the limited availability of vehicular resources. This combination creates more challenging and restrictive scenarios in the task allocation process. Next, in Figure 5b, we evaluate the impact of increasing the communication range on the task allocation rate. The results show that the allocation rate increases with the range expansion. This behavior occurs because, as explained in Figure 2b, the increased range allows for better vehicular coverage and, consequently, greater availability of vehicle resources, leading to a higher success rate in task fulfillment. Finally, in Figure 5c, we assess the allocation rate concerning the increased task weights (required resources for execution). We note that as the weight increases, the allocation percentage decreases. This decrease occurs because tasks are atomic units that are either allocated or not, and allocating sufficient resources for their fulfillment becomes more challenging as they are indivisible. ORANGE demonstrates slightly superior performance to HARMONIC in all these scenarios by 1.62%. However, no significant difference was found in any scenario when comparing the values obtained by ORANGE with the corresponding HARMONIC values (Table 4, row 1).



(a) Increasing request rate.

(b) Increasing communication range.

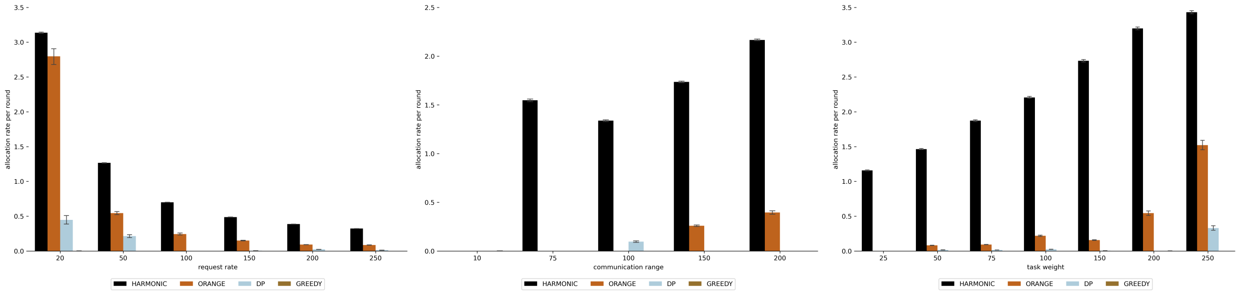
(c) Increasing task weight.

Figure 5: Allocation rate under different scenarios.

Table 4: Number of scenarios with algorithm predominance per evaluated metric

Metric	HARMONIC	ORANGE	MORFEU	GREEDY-N	DP	GREEDY
AT	0	0	0	0	0	0
TAr	247	16	0	0	8	1
TAf	84	36	17	0	0	0

In Figure 6, we evaluated the allocation rate per round (TAr) for the 272 scenarios in Table 4 (row 2) by increasing the communication range. The MORFEU and GREEDY-N algorithms did not exhibit any favorable scenarios and, therefore, do not appear in the charts. Figure 6a presents the analysis of the allocation rate per round (TAr) as it relates to the request rate for the 272 scenarios described in Table 4 (row 2). HARMONIC stands out in 90.1% (247) of the scenarios and exhibits the highest allocation rate per round among all the analyzed request rate values. HARMONIC's dominance is attributed to its approach of determining the order in which the VCs should be utilized before initiating the task allocation, unlike other algorithms prioritizing the VC with the most resources. This usage allows for greater flexibility in scenarios with different request rates. Additionally, the use of Shapley Values enables evaluating how tasks should be allocated among the VCs that provide resources in the network. As a result, HARMONIC shows allocation rates of 60.93%, 647.89%, and 127568.37% higher than ORANGE, DP, and GREEDY, respectively. The high value compared to GREEDY is due to its isolated occurrence (Table 4, row 2) with a minimum allocation percentage of 8.22×10^{-6}%. In Figure 6b, we assessed the metric concerning the communication range (CR). For $CR = 10$, GREEDY demonstrated superior performance in a single scenario, with an allocation rate per round of $8.22{\times}10^{-6}$%. No significant differences were found among the algorithms for $CR = 25$ and $CR = 50$ values. However, starting from CR = 75, HARMONIC began to stand out, and the other algorithms also appeared in some scenarios. The scenario distribution by communication range is as follows: for $CR = 75$, HARMONIC in 17 scenarios; for $CR = 100$, HARMONIC in 53 scenarios and DP in 8 scenarios; for $CR = 150$, HARMONIC in 86 scenarios and ORANGE in 7 scenarios; and for $CR = 200$, HARMONIC in 92 scenarios and ORANGE in 9 scenarios. The differentiation among the algorithms primarily occurs when $CR = 75$ due to the significant number of formed VCs and the availability of resources in the networks, as depicted in Figure 2. It is worth noting that even though a higher number of VCs are formed for $CR = 50$ (Figure 2a), there are still insufficient available resources (Figure 2b). With many formed VCs and available resources in the network, the algorithms can differentiate themselves through their allocation strategies. Therefore, from this perspective, HARMONIC exhibits higher allocation rates per round than the other solutions, surpassing ORANGE, DP, and GREEDY by 418.02%, 1659.1%, and 206280.14%, respectively. In Figure 6c, we evaluate the metric concerning the increase in task weights (TW). HARMONIC surpasses the other solutions in all considered weight values during the simulations. It is observed that as the task weights increase, the allocation rate per round also increases. This increase is because each task, when effectively allocated, directly influences the metric proportionally. Therefore, the more tasks an algorithm can allocate, the better its performance. The significant scenario distribution, considering the variation in task weight, is as follows: For $TW = 25$, HARMONIC is the only algorithm that differentiates itself in a total of 25 scenarios; for $TW = 50$, HARMONIC leads in 38 scenarios, DP in 1, and ORANGE in 3; for $TW = 100$, HARMONIC appears in 30 scenarios, DP in 3, and ORANGE in 4; for $TW = 150$, HARMONIC stands out in 18, DP in 1, and ORANGE in 2 scenarios; for $TW = 200$, HARMONIC performs better in 15 scenarios, GREEDY in 1, and ORANGE in 1; and finally, when $TW = 250$, HARMONIC appears in 10 scenarios, DP in 2, and ORANGE in 2. HARMONIC outperforms the other algorithms by 426.83%, 2831.99%, and 279085.61% compared to ORANGE, DP, and GREEDY, respectively.



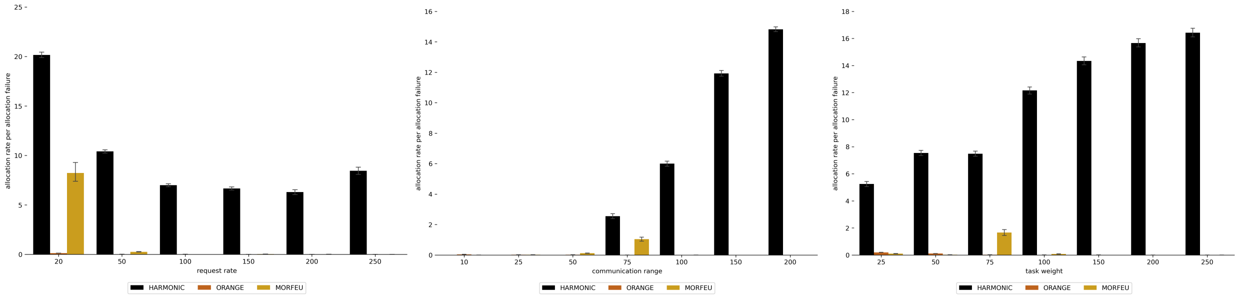
(a) Increasing request rate

(b) Increasing communication range

(c) Increasing task weight

Figure 6: Allocation rate per round under different scenarios.

Figure 7 depicts the allocation rate per allocation failure (TAf) for the 137 scenarios in Table 4 (row 3). HARMONIC outperforms in 61.31% (84) of these scenarios, while the DP, GREEDY, and GREEDY-N algorithms do not exhibit favorable outcomes. In Figure 7a, we analyze the metric concerning the request rate increase. Notably, when the request rate is 20, HARMONIC achieves its highest allocation rate per allocation failure (20.15), leveraging fewer network requests to relax the constraint and expand the solution concept space. The significant scenario distribution based on request rate variations is as follows: for $RR = 20$, HARMONIC dominates in 24 scenarios, MORFEU in 1, and ORANGE in 8; for $RR = 50$, HARMONIC outperforms in 26 scenarios, MORFEU in 2, and ORANGE in 5; for $RR = 100$, HARMONIC excels in 15 scenarios, and ORANGE leads in 10; for RR = 150, HARMONIC stands out in 10 scenarios, while MORFEU and ORANGE perform better in 6 and 3 scenarios, respectively; for $RR = 200$, HARMONIC surpasses in 4 scenarios, while MORFEU and ORANGE achieve better results in 5 scenarios each; finally, for $RR = 250$, HARMONIC prevails in 3 scenarios, alongside MORFEU and ORANGE, both in 3 scenarios. Notably, HARMONIC consistently exhibits superior metrics across all scenarios. On average, it outperforms ORANGE and MORFEU by 1189125.77% and 364945.53%, respectively. In Figure 7b, we evaluate the metric concerning the communication range (CR). For CR values of $\{10, 25, 50\}$, only the ORANGE and MORFEU algorithms stand out in 30 and 8 scenarios, respectively. However, their average allocation rate per allocation failure is 0.031%. In the best-case scenario, when the algorithms allocate approximately 1% of the requested resources, there are around 32 failures during the allocation process. For $CR = 75$, HARMONIC outperforms in 5 scenarios, ORANGE in 6, and MORFEU in 8. Despite having fewer scenarios, HARMONIC achieves higher allocation rates of 45749.19% and 3647.76% compared to ORANGE and MORFEU, respectively. From another perspective, for every 1 failure during allocation, HARMONIC allocates an average of 8.82% of the requested resources, while ORANGE and MORFEU allocate 0.02% and 0.24%, respectively. For $CR = 100$, HARMONIC has 13 favorable scenarios, while MORFEU only has 1. For CR values of 150 and 200, only HARMONIC demonstrates favorable results in 25 and 41 scenarios, respectively. Therefore, starting from $CR = 75$, our solution exhibits the highest allocation rate-per-allocation failure performance among all the compared algorithms. On average, HARMONIC outperforms the other algorithms by 45749.19% and 3647.76% compared to ORANGE and MORFEU, respectively. As explained in Figure 7a, this superiority of HARMONIC occurs when there is a higher ratio between the requested resources and the resources provided by the vehicles, making the model less restrictive and more likely to find accurate solutions. In the case of 7b, increasing the communication range makes the environment more favorable for forming more VCs due to the broader coverage of vehicles in the network, increasing the number of shared resources. Lastly, in Figure 7c, we evaluate the metric concerning the increase in task weight (TW). HARMONIC outperforms the other solutions for all the considered weight values in the evaluation process. Except for TW values of 150 and 200, where only HARMONIC and ORANGE present favorable scenarios, in all other cases, both HARMONIC, ORANGE, and MORFEU have representation. However, the number of scenarios and performances vary concerning the metric. For $TW = 25$, HARMONIC outperforms the others in 8 scenarios, ORANGE in 1, and MORFEU in 8. For $TW = 50$, HARMONIC performs better in 11 scenarios, ORANGE in 6, and MORFEU in 3. For $TW = 75$, HARMONIC performs better in 11 scenarios, ORANGE in 11, and MORFEU in 5. For $TW = 100$, there are 14 favorable scenarios for HARMONIC, 5 for ORANGE, and 4 for MORFEU. For $TW = 150$, there are 13 scenarios for HARMONIC and 6 for ORANGE. For $TW = 200$, there are 14 scenarios for HARMONIC and 2 for ORANGE. Finally, for $TW = 250$, there are 13 scenarios where HARMONIC outperforms the other algorithms, compared to 5 for ORANGE and 2 for MORFEU. Regarding the performance of the algorithms, it is evident from the chart that HARMONIC allocates more tasks per failure compared to all the compared algorithms when considering the variation of task weights in the network. HARMONIC exhibits superior performance compared to the compared algorithms by 22817.66% and 2941.37% concerning ORANGE and MORFEU, respectively.



(a) Increasing request rate

(b) Increasing communication range

(c) Increasing task weight

Figure 7: Allocation rate per allocation failure under different scenarios.

The analysis showcases that the HARMONIC algorithm efficiently allocates tasks to vehicles, resulting in fewer rounds and failures throughout the allocation process. By optimizing resource allocation and effectively distributing tasks, HARMONIC minimizes the need for additional iterations and enhances overall efficiency. It accomplishes this by identifying the optimal order of vehicle utilization and completing allocation in fewer rounds. Additionally, the algorithm's ability to decrease allocation failures contributes to the stability and reliability of the process, eliminating the need for task reassignment and minimizing disruptions. Thus, HARMONIC stands out as a more efficient and reliable option for resource allocation, allocating the same amount of resources as other methods but with fewer rounds and failures.

7. Conclusion

In this paper, we introduced HARMONIC, a heuristic algorithm that forms a coalition of vehicular clouds and leverages Shapley values to optimize resource utilization while dynamically balancing resource usage across VCs.

Simulation results demonstrate that HARMONIC successfully achieves its objectives by significantly improving resource utilization and load distribution across VCs compared to existing literature-based algorithms, considering both the number of rounds required to complete the allocation and the number of failures encountered during the process.

As described in Section 6, five of the most widely used resource allocation strategies in VC environments were simulated and compared against HARMONIC under different conditions, such as the number of resources available at each vehicle, task weight, network task load, and communication range. The methods associated with these five strategies include Game Theory, Combinatorial Optimization, Greedy Algorithms, and Dynamic Programming. The obtained results showed that HARMONIC, a Game-Theory based strategy, is the one that best responds to increasing task weight, demonstrating a better ability to distribute the workload over the available VC resources in fewer rounds and with a lower number of failures during the process. The results are summarized as follows:

- With the increasing numbers of service requests and, as a consequence, the increasing number of tasks for which resources need to be allocated, HARMONIC emerges as the best strategy, capable of generating optimized configurations for task distribution across the available VC resources. In other words, given the number of VCs and their available resources, HARMONIC consistently demonstrates the ability to utilize resources from all VCs. On the contrary, other strategies tend to concentrate resources on a subset of VCs, thereby increasing the risk of task execution delays, individual VC capacity saturation, and poor network utilization.

- As the number of resources requested by individual tasks increases while facing a limited amount of available resources, HARMONIC exhibits superior capability in satisfying the resource demands of a more significant number of requesting tasks in fewer rounds and with fewer failures during the task allocation process, thereby minimizing the need for additional resources and providing the best-balanced resource allocation.

- Considering scenarios where constraints on the number of available resources are removed, thus resulting in an abundance of resources for task allocation, HARMONIC still demonstrates the most effective utilization of resources across all VCs when compared to existing literature solutions, which often concentrate the workload on a subset of VC resources.

Additionally, as described in Section 5, HARMONIC´s task scheduling is simpler, more intuitive, and straightforward as it follows the descending order of the Shapley value attributed to each VC. In other words, it first allocates tasks to VCs with higher Shapley values.

Plans for future work include comparing HARMONIC with other existing resource allocation strategies, e.g., reinforcement learning-based strategies. We also plan to explore scenarios where forming smaller coalition structures is more beneficial than forming a bigger one. Likewise, consider other VCs' features, such as delay and network throughput beyond the available resources, to calculate Shapley values in more complex scenarios. Additionally, we plan to consider a hybrid architecture to benefit from the advantages of both centralized and decentralized architectures. Finally, we will focus on devising new mechanisms for providing incentives for creating a grand coalition to improve overall task allocation.

References

[1] R. I. Meneguette, R. E. De Grande, A. A. F. Loureiro, Intelligent Transport System in Smart Cities: Aspects and Challenges of Vehicular Networks and Cloud, Springer, 2018.

[2] A. Barbosa de Souza, P. A. Leal Rego, T. Carneiro, P. H. GonÃ§alves Rocha, J. Neuman de Souza, A context-oriented framework for computation offloading in vehicular edge computing using wave and 5g networks, Vehicular Communications 32 (2021) 100389. doi:https://doi.org/10.1016/j.vehcom.2021.100389.
URL https://www.sciencedirect.com/science/article/pii/S2214209621000589

[3] Qualcomm, Connecting vehicles to everything (2018).
URL https://www.qualcomm.com/invention/5g/cellular-v2x

[4] W. M. Danquah, D. T. Altilar, Vehicular cloud resource management, issues and challenges: A survey, IEEE Access 8 (2020) 180587–180607. doi:10.1109/ACCESS.2020.3027637.

[5] CISCO, Driving profits from connected vehicles, CISCO, White Paper (2020 [Online]).
URL https://www.cisco.com/c/en/us/solutions/service-provider/mobile-internet/driving-profits-from-connected-vehicles.html

[6] Qualcomm, Connecting vehicles to everything with c-v2x, Tech. rep. (2020).
URL https://www.qualcomm.com/research/5g/cellular-v2x

[7] L. F. Maschi, A. S. Pinto, R. I. Meneguette, A. Baldassin, Data summarization in the node by parameters (dsnp): local data fusion in an iot environment, Sensors 18 (3) (2018) 799.

[8] R. Pereira, A. Boukerche, M. A. C. da Silva, L. H. V. Nakamura, H. Freitas, G. Rocha Filho, R. I. Meneguette, Foresam - fog paradigm-based resource allocation mechanism for vehicular clouds, Sensors 21 (15).

[9] R. I. Meneguette, A. Boukerche, A cooperative and adaptive resource scheduling for vehicular cloud, in: 2017 IEEE Symposium on Computers and Communications (ISCC), 2017, pp. 398–403. doi:10.1109/ISCC.2017.8024562.

[10] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular ad hoc network, Journal of Network and Computer Applications 37 (2014) 380–392. doi:https://doi.org/10.1016/j.jnca.2013.02.036.
URL `https://www.sciencedirect.com/science/article/pii/S108480451300074X`

[11] R. I. Meneguette, R. De Grande, A. A. Loureiro, Intelligent Transport System in Smart Cities, Springer, 2018.

[12] T. Zhang, R. E. De Grande, A. Boukerche, Vehicular cloud: Stochastic analysis of computing resources in a road segment, in: Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, PE-WASUN '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 9–16. doi:10.1145/2810379.2810383.
URL `https://doi.org/10.1145/2810379.2810383`

[13] R. Meneguette, A. Boukerche, R. De Grande, SMART: an efficient resource search and management scheme for vehicular cloud-connected system, in: Proceedings of the IEEE Global Communications Conference: Mobile and Wireless Networks, Washington, USA, 2016.

[14] X. Zhang, Z. Shen, D. Yang, A permutation-based model for analysis of resource allocation overheads in vehicular ad hoc networks, IEEE Access 9 (2021) 12282–12290. doi:10.1109/ACCESS.2019.2926068.

[15] R. Zhang, L. Wu, S. Cao, N. N. Xiong, J. Li, D. Wu, C. Ma, Mpto-mt: A multi-period vehicular task offloading method in 5g hetnets, Journal of Systems Architecture 131 (2022) 102712. doi:https://doi.org/10.1016/j.sysarc.2022.102712.
URL `https://www.sciencedirect.com/science/article/pii/S138376212200203X`

[16] K. Gai, M. Qiu, H. Zhao, L. Tao, Z. Zong, Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, Journal of Network and Computer Applications 59 (2016) 46–54.

[17] R. S. Pereira, D. D. Lieira, M. A. da Silva, A. H. Pimenta, J. B. da Costa, D. Rosário, R. I. Meneguette, A novel fog-based resource allocation policy for vehicular clouds in the highway environment, in: 2019 IEEE Latin-American Conference on Communications (LATINCOM), 2019, pp. 1–6. doi:10.1109/LATINCOM48065.2019.8937912.

[18] J. Liu, J. Li, L. Zhang, F. Dai, Y. Zhang, X. Meng, J. Shen, Secure intelligent traffic light control using fog computing, Future Generation Computer Systems 78 (2018) 817 – 824. doi:https://doi.org/10.1016/j.future.2017.02.017.

[19] Q. Wu, J. Shen, B. Yong, J. Wu, F. Li, J. Wang, Q. Zhou, Smart fog based workflow for traffic control networks, Future Generation Computer Systems 97 (2019) 825 – 835. doi:https://doi.org/10.1016/j.future.2019.02.058.

[20] A. Paul, N. Chilamkurti, A. Daniel, S. Rho, Intelligent vehicular networks and communications: fundamentals, architectures and solutions, Elsevier, 2016.

[21] R. Meneguette, R. De Grande, J. Ueyama, G. P. R. Filho, E. Madeira, Vehicular edge computing: Architecture, resource management, security, and challenges, ACM Computing Surveys (CSUR) 55 (1) (2021) 1–46. doi:10.1145/3485129.
URL `https://doi.org/10.1145/3485129`

[22] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, A. Mouzakitis, A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles, IEEE Transactions on Intelligent Transportation Systems 23 (7) (2022) 6206–6221. doi:10.1109/TITS.2021.3084396.

[23] W. Yang, R. Zhang, C. Chen, X. Cheng, Secrecy-based resource allocation for vehicular communication networks with outdated csi, in: 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), IEEE, 2017, pp. 1–5.

[24] D. D. Lieira, M. S. Quessada, A. L. Cristiani, R. I. Meneguette, Resource Allocation Technique for Edge Computing Using Grey Wolf Optimization Algorithm, in: 2020 IEEE Latin-American Conference on Communications (LATINCOM), 2020, pp. 1–6.

[25] J. B. D. da Costa, M. L. M. Peixoto, R. I. Meneguette, D. L. Rosário, L. A. Villas, MORFEU: Mecanismo baseado em Otimização Combinatória para Alocação de Tarefas em Nuvens Veiculares, in: Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), SBC, 2020, pp. 505–518. doi:10.5753/sbrc.2020.12305.

[26] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, S. Wan, Multi-objective optimization for resource allocation in vehicular cloud computing networks, IEEE Transactions on Intelligent Transportation Systems 23 (12) (2022) 25536–25545. doi:10.1109/TITS.2021.3091321.

[27] H. Peng, X. Shen, Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks, IEEE Transactions on Network Science and Engineering 7 (4) (2020) 2416–2428.

[28] S. Zamir, M. Maschler, E. Solan, Game theory, Cambridge University Press, 2020.

[29] Z. Sun, Y. Liu, J. Wang, G. Li, C. Anil, K. Li, X. Guo, G. Sun, D. Tian, D. Cao, Applications of Game Theory in Vehicular Networks: A Survey 23 (4) (2021) 2660–2710. doi:10.1109/COMST.2021.3108466.

[30] A. Safdarian, P. H. Divshali, M. Baranauskas, A. Keski-Koukkari, A. Kulmala, Coalitional game theory based value sharing in energy communities, IEEE Access 9 (2021) 78266–78275.

[31] O. N. Bondareva, Some applications of linear programming methods to the theory of cooperative games, Problemy kibernetiki 10 (119) (1963) 139.

[32] L. S. Shapley, On balanced sets and cores, Tech. rep., The RAND Corporation (1965).

[33] L. S. Shapley, A value for n-person games, Contributions to the Theory of Games (AM-28) - Princeton University Press 2 (1953) 307–317.

[34] R. Mitchell, J. Cooper, E. Frank, G. Holmes, Sampling permutations for shapley value estimation, Journal of Machine Learning Research 23 (43) (2022) 1–46.

[35] E. Štrumbelj, I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowledge and information systems 41 (2014) 647–665.

[36] Q. Luo, C. Li, T. Luan, W. Shi, Minimizing the delay and cost of computation offloading for vehicular edge computing, IEEE Transactions on Services Computing 1374 (2021) 1–12.

[37] G. Hattab, S. Ucar, T. Higuchi, O. Altintas, F. Dressler, D. Cabric, Optimized assignment of computational tasks in vehicular micro clouds,

in: Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking (EdgeSys 2019), ACM, 2019, pp. 1–6.

[38] X. Wu, S. Zhao, R. Zhang, L. Yang, Mobility prediction-based joint task assignment and resource allocation in vehicular fog computing, in: IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2020, pp. 1–6.

[39] R. S. Pereira, D. D. Lieira, M. A. C. da Silva, A. H. M. Pimenta, J. B. D. da Costa, D. Rosário, L. Villas, R. I. Meneguette, RELIABLE: Resource allocation mechanism for 5g network using mobile edge computing, Sensors 20 (19) (2020) 5449. doi:10.3390/s20195449.
URL `https://doi.org/10.3390/s20195449`

[40] J. B. D. da Costa, R. I. Meneguette, D. Rosário, L. A. Villas, Combinatorial optimization-based task allocation mechanism for vehicular clouds, in: Proceedings of the IEEE 91st Vehicular Technology Conference (VTC Spring), IEEE, 2020, pp. 1–5.

[41] S.-S. Lee, S. Lee, Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information, IEEE Internet of Things Journal 7 (10) (2020) 10450–10464. doi:10.1109/JIOT.2020.2996213.

[42] C. Tang, C. Zhu, X. Wei, H. Wu, Q. Li, J. J. Rodrigues, Intelligent resource allocation for utility optimization in rsu-empowered vehicular network, IEEE Access 8 (2020) 94453–94462.

[43] H. A. P. Marques, R. I. Meneguette, Um Mecanismo de Alocação de Recursos em Nuvens Veiculares baseado em Teoria dos Jogos, in: Anais Estendidos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), SBC, 2021, pp. 241–248.

[44] R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, Cooperative resource management in cloud-enabled vehicular networks, IEEE Transactions on industrial electronics 62 (12) (2015) 7938–7951.

[45] A. Ribeiro Jr., J. Costa, G. R. Filho, L. Villas, D. Guidoni, R. Meneguette, Alocação de tarefas em nuvens veiculares utilizando jogos de mercado, in: Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, SBC, Porto Alegre, RS, Brasil, 2022, pp. 210–223. doi:10.5753/sbrc.2022.222247.
URL `https://sol.sbc.org.br/index.php/sbrc/article/view/21172`

[46] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using sumo, in: 2018 21st international conference on intelligent transportation systems (ITSC), IEEE, 2018, pp. 2575–2582.

[47] S. Uppoor, M. Fiore, Large-scale urban vehicular mobility for networking research, in: 2011 IEEE Vehicular Networking Conference (VNC), 2011, pp. 62–69. doi:10.1109/VNC.2011.6117125.

[48] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, 1996, p. 226–231.

[49] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the american statistical association 32 (200) (1937) 675–701.

[50] P. B. Nemenyi, Distribution-free multiple comparisons., Princeton University, 1963.

[51] O. J. Dunn, Multiple comparisons using rank sums, Technometrics 6 (3) (1964) 241–252.

[52] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (1) (2006) 1–30.
URL `http://jmlr.org/papers/v7/demsar06a.html`

[53] M. Nabi, R. Benkoczi, S. Abdelhamid, H. S. Hassanein, Resource assignment in vehicular clouds, in: 2017 IEEE International Conference on Communications (ICC), IEEE, 2017, pp. 1–6.