



A survey on computation resource allocation in IoT enabled vehicular edge computing

Naren¹ · Abhishek Kumar Gaurav¹ · Nishad Sahu¹ · Abhinash Prasad Dash¹ · G. S. S. Chalapathi¹ · Vinay Chamola^{1,2}

Received: 24 April 2021 / Accepted: 19 July 2021
© The Author(s) 2021

Abstract

The number of vehicles is increasing at a very high rate throughout the globe. It reached 1 billion in 2010, in 2020 it was around 1.5 billion and experts say this could reach up to 2–2.5 billion by 2050. A large part of these vehicles will be electrically driven and connected to a vehicular network. Rapid advancements in vehicular technology and communications have led to the evolution of vehicular edge computing (VEC). Computation resource allocation is a vehicular network's primary operations as vehicles have limited onboard computation. Different resource allocation schemes in VEC operate in different environments such as cloud computing, artificial intelligence, blockchain, software defined networks and require specific network performance characteristics for their operations to achieve maximum efficiency. At present, researchers have proposed numerous computation resource allocation schemes which optimize parameters such as power consumption, network stability, quality of service (QoS), etc. These schemes are based on widely used optimization and mathematical models such as the Markov process, Shannon's law, etc. So, there is a need to present an organized overview of these schemes to help in the future research of the same. In this paper, we classify state-of-the-art computation resource allocation schemes based on three criteria: (1) Their optimization goal, (2) Mathematical models/algorithms used, and (3) Major technologies involved. We also identify and discuss current issues in computation resource allocation in VEC and mention the future research directions.

Keywords Resource allocation · Vehicular edge computing (VEC) · Mobile edge computing (MEC) · IoV

Introduction

Rapid strides made in networking technologies have catapulted the number of devices connected to each other over the internet. It has led to the advent of the internet of things (IoT) [1]. IoT has not just led to an increase in the number of networked devices but has made its influence on other domains like industries through industrial internet of things (IIoT) [2], vehicular networks through the internet of vehicles (IoV) [3,4]. IoT provides a potential economic value of 150–500 billion per year. With the IoT enabled VEC, the computational capacity in the vehicular application increases. Different computing paradigms are used in IoT to provide different services for diverse applications. ITS (intelligent transport system) plays an important role in improving the road safety and managing the traffic. VANET (vehicular ad hoc network) is a key enabler in ITS. Communication in VANETs is realized by V2V (vehicle to vehicle), V2R (vehicle to RSU) and V2X (vehicle to everything). IoV

✉ Vinay Chamola
vinay.chamola@pilani.bits-pilani.ac.in

Naren
f2015547@pilani.bits-pilani.ac.in;
h20150547@pilani.bits-pilani.ac.in

Abhishek Kumar Gaurav
h20200185@pilani.bits-pilani.ac.in

Nishad Sahu
h20160215@pilani.bits-pilani.ac.in

Abhinash Prasad Dash
h20200182@pilani.bits-pilani.ac.in

G. S. S. Chalapathi
gssc@pilani.bits-pilani.ac.in

¹ Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus, Pilani 333031, India

² APPCAIR, BITS-Pilani, Pilani Campus, Pilani, India

(internet of vehicles) has facilitated the use of smart vehicles, which are used in various applications like data storage and processing. IoT has extended its paradigm to vehicular edge computing with the help of technologies such as SDN, AI, blockchain, cloud computing, etc. With the help of these technologies, billions of connected vehicle devices in the IoT run directly at the network edge. Many companies have heavily invested in these technologies for various IoV applications such as street monitoring, real-time navigation, autonomous and other safety-related systems. In of the projects of Google, they have used AI for autonomous driving. They have used images, ultrasound, GPS navigation, beams of radars and central computing to identify objects and predict behavior accurately [5–7].

IoT has also led to the advent of multiple domains in mobile computing like fog/edge computing [8,9], mobile edge computing [10], vehicular edge computing (VEC) [11] among others. Mobile devices like smartphones, tablets, and wearable devices are witnessing massive growth in the varieties of applications supported by them and the number of users worldwide. The current network devices or servers (generally cloud servers which cater to applications for which the user devices fall short in terms of storage/computation or network resources) will come under immense load while serving the needs of these devices, which are rapidly increasing in numbers. Vehicular edge computing (VEC) is being explored as a promising paradigm to reduce stress and congestion on the core network by providing the necessary computation and data storage resources. This method of reducing the load on network devices or servers is quite similar to the concept of mobile edge computing (MEC), except that MEC uses stationary edge devices such as road side units (RSUs) or Cloudlets. In contrast, VEC uses vehicles as edge devices [12].

The vehicles run different applications such as infotainment, safe-driving assistance, navigation, autonomous driving, etc which are in focus due to emerging scenarios such as vehicular ad hoc networks (VANETs) [13] and vehicle-2-everything (V2X) [14]. But all of these applications need computation, storage, and network resources. It has been estimated that a typical autonomous vehicle (AV) could generate up to 1GB of data per second [15]. These vehicles themselves rely on computation offloading (CO) to the cloud servers for many computationally intensive tasks. It can be expected that as the number of AVs on the roads increases, these vehicles themselves increase the load on the cloud servers and cause congestion in the network. In addition, this CO by the AVs to the cloud servers is much more complex when compared to CO by non-mobile network entities due to the high mobility and geographically distributed nature of the AVs. The cloud servers may be located at faraway locations, leading to increased latency and reduced reliability of these connections [16].

Therefore, a vehicle may simultaneously function in two roles: as a task vehicle (TaV), which requests services from cloud servers and also as a service vehicle (SeV), which assists nearby resource-constrained devices [17]. Hence, in order to achieve fast and efficient computation for the vehicles and nearby resource-constrained devices while simultaneously minimizing the load on the cloud services and the communication network is a challenging and multi-faced resource allocation problem in VEC.

Figure 1 gives an overview of the work done in this survey section by section. The rest of this paper is organized as follows. The next section provides a preliminary background on VEC and CRA strategies in VEC. The following section provides a literature review of related surveys. The next section gives an overview of this survey, while the following sections present classifications of CRA schemes for VEC in terms of the optimization goal of RA, the underlying technology used, mathematical models/algorithms used, and significant technologies used in the scheme, respectively. We list open research issues and future research directions in the next section and finally conclude our paper in the last section.

Preliminary background

Vehicular edge computing architecture

Based on the works studied in this survey, we present in Fig. 2, a generalized VEC architecture to capture the necessary connectivity, communication and mobility aspects in order to understand and analyze the different resource allocation problems and solutions that have been proposed in the literature.

As can be seen in Fig. 2, a typical VEC architecture includes the primary task entities (PTEs)—resource-constrained devices such as smartphones, tablets and laptops of pedestrians and the passengers of a vehicle. Depending on the scenario, the computationally intensive tasks running on these devices could be offloaded to either parked vehicles or even moving vehicles in case the connection is deemed stable. The vehicles in this VEC could be moving/parked non-autonomous/ autonomous cars, buses, trucks, etc. These vehicles may/may not have the capability to function as SeVs or TaVs or both simultaneously. TaVs may offload their tasks to edge servers located at cellular base stations or RSUs, which may further offload to cloud servers based on the requirement. Additionally, it might be possible that different vehicles have different computational capabilities (for example, an ordinary vehicle has much fewer resources than an autonomous/smart vehicle). Hence the possibility of inter-vehicle offloading also exists [18].

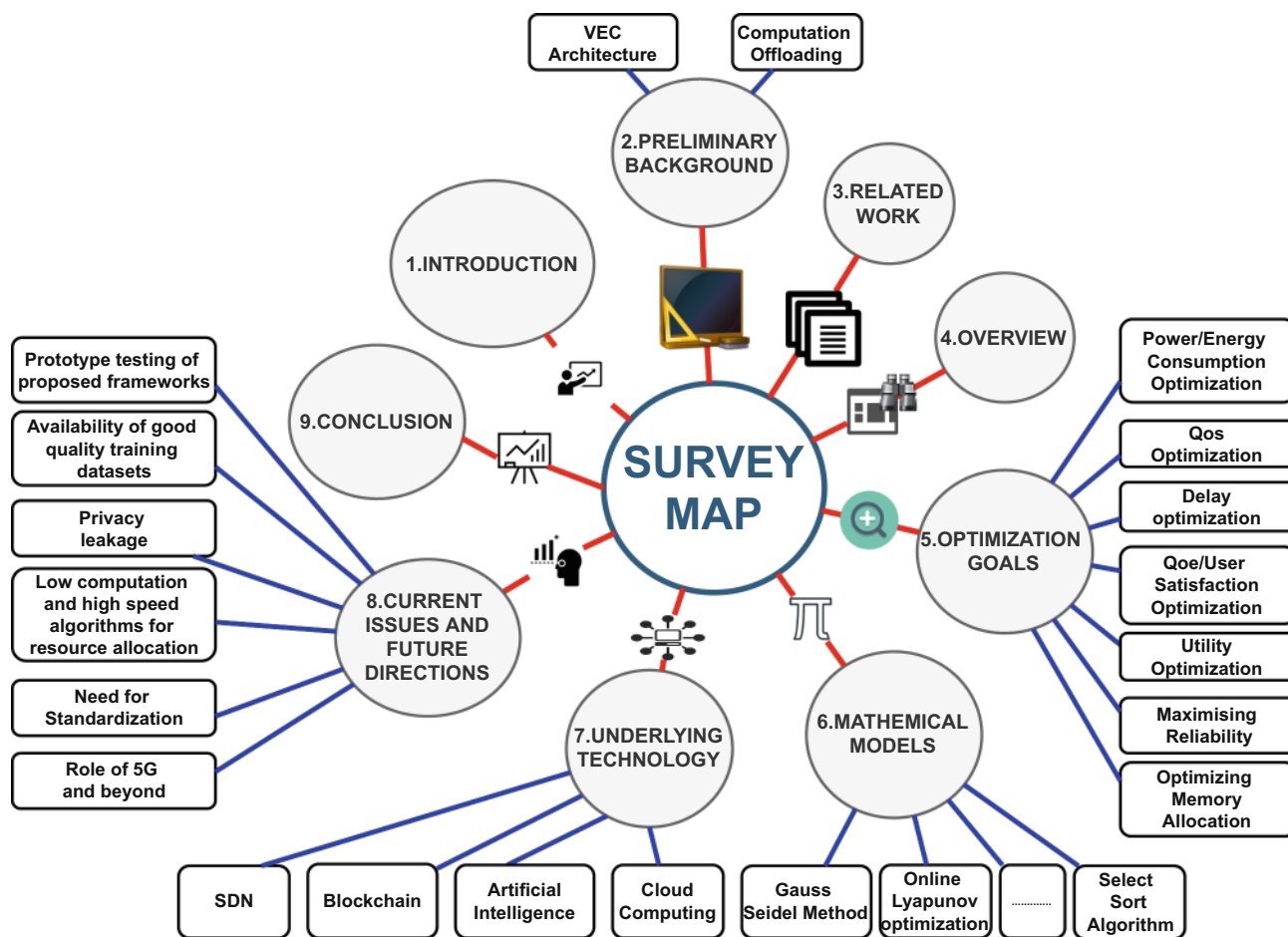


Fig. 1 Map of this survey showing various sections and classification criteria

Computation offloading

A typical computation offloading happens in the following manner. First, a resource-constrained device (either TaV or PTE) will identify a suitable node on the network to offload its computation—this could be an SeV, RSU/cellular edge servers or a cloud server [19]. Intermediate entities on the network may assist/supervise the process of identifying a suitable offloading destination which could be either one or multiple hops away on the network. Once the offloading destination is determined, the tasks to be offloaded are communicated to the destination using a suitable communication medium. This could be Wi-Fi/Cellular/DSRC or a combination of these. The offloaded tasks are scheduled and executed at the destination nodes, and the results are communicated back to the respective devices.

We take the example of [20] to explain this process. The authors of [20] propose a reliable computation offloading scheme for a network with fixed and mobile edge computing nodes (vehicles). In their scheme (Fig. 3), we can see that a vehicle executes some tasks locally and offloads par-

tially. It first sends a service request to a nearby Access RSU. This RSU communicates with an SDN controller running a suitable algorithm to obtain details about the offloading decision. Based on the decision, the vehicle communicates the task to be offloaded to the RSU. Based on the computational requirements of the offloaded task, the RSU may further partition it into independent sub-tasks and allocate it among single/multiple fixed and mobile edge nodes for parallel execution. Once the edge nodes finish their executions, they send the results to the. The RSU merges these individual results and sends them back to the vehicle.

Table 1 lists the abbreviations used in this paper.

Related work

While there are several works that have surveyed resource allocation in vehicular networks or edge networks per say, at present there are very few works that have surveyed resource allocation or computation resource allocation schemes in vehicular edge computing.

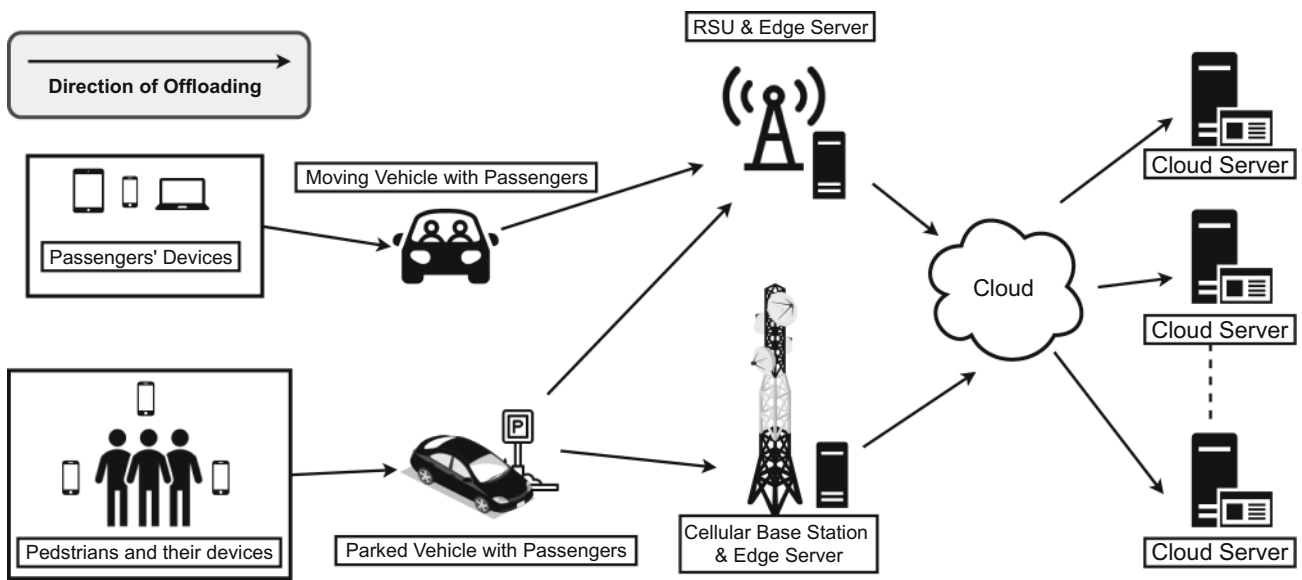


Fig. 2 General VEC architecture

Fig. 3 An offloading example. . Adapted from [20]

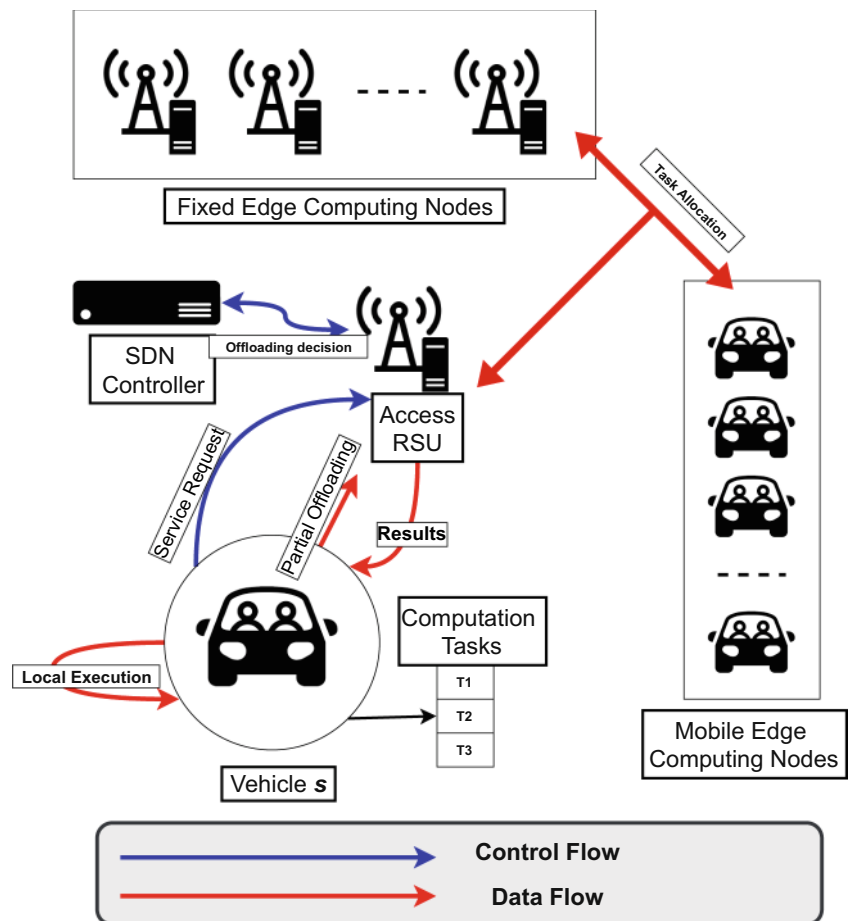


Table 1 List of abbreviations

Notation	Description
VEC	Vehicular edge computing
MEC	Mobile edge computing
IoV	Internet of vehicles
IoT	Internet of things
V2X	Vehicle to everything
UE	User equipment
RA	Resource allocation
RSU	Road side unit
VECC	Vehicular edge cloud computing
CAVs	Connected autonomous vehicles
SDN	Software defined network
DRL	Deep reinforcement learning
AI	Artificial intelligence
VANET	Vehicular adhoc network
TaV	Task vehicle
SeV	Service vehicle
PTE	Primary task entity
CO	Primary computation offloading
QoS	Quality of service
NFV	Network function virtualization

Noor-A-Rahim et al. [21] have surveyed and classified RA schemes based on the type of network they are used in DSRC, Cellular and heterogeneous vehicular networks. Zheng et al. [22] present a survey on computational offloading in edge computing based on different offloading scenarios among edge servers, IoT devices, and cloud servers), factors (device, network, service and user factors) and strategies (full and partial offloading). They also discuss program partitioning and the efficient selection of program components to offload.

Jiang et al. [23] discuss several aspects of computation offloading, such as minimizing energy consumption, enhancing and guaranteeing quality of service and user experience. They discuss the general problems of what to offload, where to offload, and when to offload and a few case studies based on game theoretic and heuristic based computation offloading. Lin et al. [24] identify the need and usage of resource allocation in the industrial internet of things (IIoT). They list a few works of resource allocation developed for different environments such as VEC, cloud, software defined networks, mobile edge computing, etc., which could be utilized for IIoT applications.

References [12,16] are the only works that strictly deal with RA in VEC. Dziyauddin et al. [16] identify that although mobile data offloading mechanisms have received significant attention, VEC has failed to receive much attention from the research community. They identify that content caching and delivery (CachDel), and CO are significant resource allocation

challenges to be addressed in vehicular edge networks and carry out a thorough survey on the optimization techniques which have been published to address these problems. Their classification of CO schemes is based on single/hybrid and multi optimization approaches. Within each category, they classify works based on single/hybrid/multiple optimization goals such as QoS, energy efficiency and cost improvement.

Boukerche and Soto [12] present a survey on computation offloading in VEC. They discuss the algorithms, models and classification in computation offloading and retrieval. They provide a detailed review of the works on computational offloading based on the three steps involved: partitioning, scheduling, and data retrieval. However, a significant amount of work on resource allocation in VEC has been carried out after [12] and hence these works remain to be surveyed. Therefore our work serves as both an initial survey on RA for VEC and it also serves as a compliment by including the latest works on CRA.

Hence, to the best of our knowledge, resource allocation in VEC has not received much attention in contemporary literature. Thus, there is a requirement for a thorough and comprehensive survey of resource allocation schemes explicitly designed for VEC. The very few works that do survey research in this area have not covered most of the novel research which has been published in the past 2–3 years. Broadly, resource allocation in VEC comprises of computation resource allocation, memory resource allocation and network resource allocation. In this work, we conduct an extensive survey of various computation resource allocation schemes designed explicitly for VEC and present our findings in a comprehensive manner to clearly identify the developments, breakthroughs and open research issues in this field. Memory and network resource allocation are not covered in this survey. The major contributions of our work are listed below.

- 1 An extensive and comprehensive survey of state-of-the-art computation resource allocation (CRA) schemes for VEC.
- 2 Classification of CRA schemes for VEC based on their goals, namely optimizing parameters like power/energy consumption, QoS, delay, QoE, utility, reliability and memory allocation.
- 3 Classification of CRA schemes for VEC based on algorithms and mathematical models used in their framework and problem formulation.
- 4 Classification of CRA schemes for VEC based on underlying technology like SDN, blockchain, AI and cloud computing.
- 5 Identification of open research issues and future research direction in vehicular edge computing to drive forward

research and development towards the advancement of vehicular networks and applications.

In summary, [16]’s classification of CRA in VEC is based on single/hybrid/multi optimizations, while [12]’s study is based on the different stages of the computational offloading process. In this work, we adopt a different perspective by studying and classifying the latest works on CRA in VEC based on their optimization goals, the mathematical formulation/models used in the schemes, and their underlying technologies. Moreover, this work serves as a much-needed review to cover the latest works on CRA in VEC, which have been published after [12,16].

Computational offloading in IoT-enabled VEC—survey overview

Edge computing enabled software-defined IoV (Ec-SDIoV) enhances the capability of IoV, which improves the latency services and reduces the computational complexities. The Ec-SDIoV architecture facilitates the RSUs (roadside units), mobile vehicles for reducing the computational complexity co-operatively via SDN [20]. Vehicular networks enabled with IoT technology have become an essential part of the ITS (intelligent transport system). It includes various applications for passengers and vehicles, such as online gaming, augmented reality (AR) for passengers, autonomous driving, and intelligent auxiliary driving for vehicles. These applications require intensive computational and communication resources [25]. IoV combined with various other technologies helps in improving the efficiency of vehicular applications. In VEC-based IoT network, the devices can offload all or part of the computational tasks to the VEC server, which reduces the latency and save energy for devices [26]. With respect to the IoT services scenario, the computational complexity reduces in VEC and enhances the energy consumption. The growth of IoT has led to the evolution of VANETs. The IoT does not support traditional VANET services and improves the various applications by introducing various computing and technological paradigms.

There are many research works on resource allocation in vehicular edge computing. To understand this area, it is crucial to classify these studies under broad categories, which may clarify where improvements can be made in its development. Keeping this idea in mind and after an extensive literature survey, the schemes available for VEC can be classified based on the following criteria.

- 1 Optimization goals in resource allocation—power/energy optimization, delay optimization, quality of ser-

vice (QoS) optimization, utility optimization, user experience optimization, etc.

- 2 Underlying technology—cloud computing, software defined networks, blockchain, machine learning, reinforcement learning and deep learning.
- 3 Mathematical models and algorithms used.

We shall explore these classifications in Sects. “Classification of RA frameworks in VEC based on optimization goals for IoT applications”, “Classification of RA frameworks in VEC based on mathematical and computational models/algorithms/techniques used” and “Classification of RA frameworks in VEC based on underlying technologies” respectively. The summary of the survey has been presented Table 5.

Classification of RA frameworks in VEC based on optimization goals for IoT applications

IoT applications require low power/energy consumption, less latency and low computation and memory resource consumption. Low power/energy consumption ensures that the system can be implemented on IoT edge servers, mobile platforms like vehicles, etc, and can be deployed for a long-term application. Less latency ensures that the network is more secure and gives less opportunity to adversaries to carry out attacks. Moreover, vehicles move at high speeds. Hence, it is necessary to complete the resource allocation fast so that the driver does not have to explicitly wait near RSUs or other edge devices to offload its computational work and be done even when the vehicle is running at a reasonable speed. Also, vehicles have limited on board computation and memory resources, so it is essential to ensure that the framework uses the least possible resources.

One of the important resource allocation issues is the parameter we want to optimize while allocating the resources. So CRA frameworks designed for VEC need to ensure that these parameters are optimized so that they can be deployed in IoT scenarios. Based on this, the algorithm for the resource allocation may vary. Additionally, apart from being feasible for IoT deployment, optimizing other parameters like QoS, Utility, reliability, etc are essential for ensuring that the system achieves its intended functionality. Thus the parameter which is optimized by the framework is an essential parameter for classification and based on optimization goals, we classify the frameworks into the following categories (summarized in Table 2).

Power/energy consumption optimization

Power/energy is one of the main factors in vehicular networks as vehicles have limited battery capacity and storage.

Table 2 Classification of frameworks according to their optimization goals

Optimization Goal	References
Power/energy	[28–30]
QoS	[33–35]
Delay	[25,26,28,35–39]
QoE and user satisfaction	[34,37]
Network/system utility	[26,33,37,39,40]
Reliability	[20]

However, at the same time, they are required to perform many intensive computations as part of the network [27]. Hence, authors in [28–31] try to optimize energy and power consumption in their resource allocation schemes. Optimizing power includes various factors like transmission power, uploading time, offloading ratio, local CPU frequency, etc and all these factors need to be considered while formulating the energy/power optimization problem.

In [29], an online Lyapunov optimization process is followed to find a solution to the problem of computation resource allocation to get a trade-off between the mean weighted power consumption and delay. Their simulation results showed that their joint computation offloading and URLLC resource allocation for collaborative MEC-assisted cellular-V2X networks could improve trade-off performance compared to centralized MEC-assisted V2X. In [28], the authors formulated a problem based on binary optimization problem to minimize the entire cost of the entire system in terms of energy and time and guarantee efficient resource utilization among multiple vehicles which are part of the network [30].

QoS optimization

The term QoS was coined by the International Telecommunication Union (ITU) in 1994 [32]. It refers to network connection requirements like response time, signal-to-noise ratio, cross-talk interrupts, etc. In short, it is actually an overall performance metric for a computer network, cloud network, etc., more specifically, the performance from the point of view of the users of the network. It is also very close in meaning to user satisfaction. As in any network, revenue generation is directly dependent on the network users, and it is crucial to have a good QoS. In works [33–35] focus on maximizing their QoS in their resource allocation schemes for VEC networks.

In [35], the authors adopt a deep Q-learning based approach in their MEC-based IoV system for optimized RA while simultaneously maintaining QoS. In [34] the authors systematically look into the QoS guaranteed optimal RA problem for connected vehicular fleets in MEC environ-

ments. To demonstrate the working of their framework, they take two large real-world datasets of online-taxi services and evaluate their QoS with their framework. Results show that their proposed method saved up to 40 percent costs compared to naive resource provisioning strategies.

Delay optimization

Delay optimization or reducing the network latency is an essential goal in vehicular networks as they are of very high speed in nature. Works [25,26,28,35–39] focus in reducing their delay while implementing their resource allocation scheme. Ensuring less delay makes it possible to run latency-sensitive applications like assistive driving [20].

As mentioned previously [26], use Lyapunov optimization to optimize both delay and energy jointly and [38] formulate their offloading problems as a constrained optimization problem where the aim is to optimize the delay. In [40], the author formulates a joint optimization load balancing and offloading problem in the form of a system utility maximization problem under a permissible latency constraint.

QoE/user satisfaction optimization

Ensuring high QoE (quality of experience) or user satisfaction is significant for the networks to ensure adequate participation from users. Studies [34,37] focus on ensuring a good QoE and user satisfaction while implementing their resource allocation frameworks.

Utility optimization

While focusing on optimizing parameters like speed, energy, etc, it is equally important to optimize the utility of the entire network as a whole and ensure all the resources are adequately utilized. Hence network and system utilization is also a very important issue. Works [26,33,37,40] implement resource allocation schemes where they try to optimize utility or at least ensure it is above a certain threshold value.

In [39], while modeling a system based in IOV based on edge computing, the main objective is to optimize the latency and ensure maximum utilization of resources has been done. The minimal latency and maximum resource utilization rate are calculated by offloading the computing task. To increase the maximum utilization of resources, the NSG-III method has been adopted to find the optimal global solution for the computation offloading method. This NSG-III is a non dominated sorting genetic algorithm.

Maximizing reliability

In [5], the authors have proposed to use the resources available of nearby RSUs and vehicles to facilitate the func-

functionalities of edge computing keeping the constraints of latency and computationally-intensive requests in mind. In a dynamic and complex IoV environment, the interruption of both communication links and processing nodes becomes unavoidable, which may lead to adverse conditions. To ensure timely completion of delay-sensitive IoV services, the authors proposed a partial CO and reliable task allocation scheme which supports a reprocessing method. A non-convex optimization problem and a fault-tolerant particle swarm optimization algorithm is employed to ensure task completion before the deadline.

Optimizing memory allocation

In [20], the authors have proposed to use the resources available of nearby RSUs and vehicles to facilitate the functionalities of edge computing, keeping the constraints of latency and computationally intensive requests in mind. In a dynamic and complex IoV environment, the interruption of both communication links and processing nodes becomes unavoidable, which may lead to adverse conditions. To ensure timely completion of delay-sensitive IoV services, the authors proposed a partial CO and reliable task allocation scheme which supports a reprocessing method. A non-convex optimization problem and a fault-tolerant particle swarm optimization algorithm are employed to ensure task completion before the deadline.

Summary of this section

The issue that arises with the resource allocation is the question of the parameter to be optimized. As a result, the algorithms vary and the frameworks used are classified. Some of them are power energy consumption optimization, QoS optimization, delay optimization, QoE/user satisfaction optimization, utility optimization, maximizing reliability and optimizing memory allocation. It is vital to consider transmission power, uploading time, offloading ratio, local CPU frequency, etc. to formulate the energy/power optimization problem. An online Lyapunov optimization process, Binary optimization problem gives insights into this. The second one is the overall performance metric for a computer network, cloud network, etc. from the user point of view. Deep Q-learning based approach and considering the QoS guaranteed optimal RA problem helped to achieve that. Lyapunov optimization and a joint optimization load balancing and offloading problem help demonstrate problems based on reducing the network latency. Ensuring user satisfaction and optimizing the utility of the entire network is also an essential parameter.

Classification of RA frameworks in VEC based on mathematical and computational models/algorithms/techniques used

Performing resource allocation among various entities involved in a vehicular edge computing scenario is a complex task where many parameters are involved. Hence most studies adopt well-established mathematical and computational techniques to model the resource allocation problem and solve it by optimizing the required parameters. Thus, it is important to discuss various mathematical and computational models/algorithms/techniques. So, in this section, we briefly discuss these as given below.

Gauss Seidel method

The Gauss Siedel method or the method of successive displacement or Liebmann method is a type of iterative method employed to solve a system of linear equations. It is similar to the Jacobi method and is named after German mathematicians Carl Friedrich Gauss and Philipp Lied von Siedel. Interestingly it was only present in a private letter from Gauss to one of his students, Gerling, in the year 1823, and a publication on this method was not present until the year 1874 [41,42].

In [29], the authors use the Gauss–Seidel method for calculating the processor frequency to find power consumed for running the Lyapunov Optimization method to solve computation resource allocation problem in a collaborative MEC assisted cellular-V2X network. The following equation can give the power dissipated via the output pins of a CPU core:

$$P = 0.5 \times C \times V^2 \times f, \quad (1)$$

where P is the power consumed by the processor, C is the capacitance, V is the supply voltage, and f is the operating frequency f the processor core. So, the term f is calculated using the Gauss Siedel method.

Online Lyapunov optimization

Lyapunov optimization is the use of the Lyapunov function to ensure the stability of the dynamic system. In [29], the authors have proposed a Lyapunov Optimization Technique which solves the problem of resource allocation to get a trade-off between the energy consumption and the delay. Also, in [36], the authors have used this technique to solve the ARAEUE algorithm, which was designed to enhance the user experience by minimizing the quality loss without failing to meet the deadline.

Table 3 Various mathematical algorithms and models used in the frameworks

Mathematical/computational models and algorithms	References	Advantages	Disadvantages
Gauss Seidel method	[29]	Used for finding frequency for calculating the power	Makes the programming of the network difficult as each iteration requires more computation time per iteration
Online Lyapunov optimization	[29,36]	Enhances the user experience by providing the trade off between energy consumption and delay	There is no general method available to construct Lyapunov functions
Game theory	[26,29]	Reduces the inter-cell interference, increases the throughput and guarantees the reliability of URLLC V2X communication by choosing the best offloading decisions	Sometimes the assumptions can be unrealistic while making certain offloading decisions which may not produce the best outcome
T-slot look ahead algorithm	[36]	Reduces the computing quality loss and provides a computationally efficient solution	Energy consuming
Bipartite graph	[29,40]	Solves the NP-hard energy optimization problem when all the vehicles offload their tasks to the same VEC server	Solve the energy optimization problem but does not solve the latency issues
Heuristic algorithm	[20,47]	Used for reliability maximization and solves the problem of allocating resources to fog computing in vehicular applications.	Often used only when approximate solutions are sufficient and exact solutions are not necessary
Markov model	[26,35,37]	Reduces the computational limitations and delay of the VEC	Markov Models are generally inappropriate over sufficiently short intervals
Fault-tolerant particle swarm optimization (FPSO-MR)	[20]	Converts non-convex functions into convex functions for to solve the formulated problem with less convexity	It has a very low convergence rate in the iterative process.
Convex optimization	[36]	Replaces the non-convex functions and restrictions by acceptable convex functions which helps in reducing the energy consumption	Difficult to design a suitable model and develop efficient, fast, scalable and distributed algorithm to solve large practical issues
Reinforcement learning	[28]	Optimizes the computation offloading scheduling problem by which energy and delay are minimized in long term	Needs a lot of data and requires extensive computation which may sometimes diminishes the result
Newton–Raphson method	[25]	Reduces the complexity in case of huge overheads using convex optimization algorithm	Sometimes the convergence of Newton–Raphson method is not guaranteed especially in case of multiple roots

Table 3 continued

Mathematical/computational models and algorithms	References	Advantages	Disadvantages
Distributed deep-Q learning	[28]	Minimizes the overall computational and energy consumption	Requires a large amount of data for better performance and also it is overly sensitive to the changes in the data set
Non-dominated sorting genetic algorithm	[39]	Improves the user experience and reduces the latency	Genetic algorithms are non-deterministic method. Thus, it may give different solution each time the algorithm is run for same instance which makes it non-optimal in some cases
Multiple criteria decision making	[39]	Provides the most optimal solution for the computational offloading strategies	Time consuming
Select sort algorithm	[34]	Allocates the minimum required resources which helps in reducing the energy consumption	This algorithm is beneficial when used for small data but its efficiency decreases when dealing with large amount of data
Randomly select algorithm	[34]	Uses random resources for allocation based on game theory which helps in reducing the latency	There is very little control on what the model does and sometimes its efficiency is not optimal as it does random selection

Game theory

Game theory is an approach where the interests of many players are maximized under some given constraints. During the game, the players rely on their information and formulate a strategy for optimally playing the game. A game consists of three things typically: (1) payoff function, (2) the strategy, and (2) the players. Players can be an individual or a team who has the freedom to decide his own strategy and arrive at his judgment. Game models are divided into two categories: (1) Classical games and (2) Evolutionary games. Classical games can be classified as non-cooperative games or cooperative games, and into dynamic games or static games depending on the criteria, we choose for classification. [43,44]

Cooperative and non-cooperative games

The non-cooperative and cooperative games are differentiated on the basis of the fact that whether the players participating in them reach a cooperative relationship during the course of the game or not. If yes, then it is a cooperative game otherwise, it is a non-cooperative game. The non-operative game is majorly based on the Nash equilibrium concept. In [29], the authors use the non-cooperative game and bipartite graph to reduce the inter-cell interference and do

channel allocation maximizing the throughput while guaranteeing the reliability of URLLC V2X communication in their proposed joint computation offloading and URLLC resource allocation frameworks for collaborative MEC assisted cellular V2X networks.

Static and dynamic games

The dynamic and static games are classified based on the order in which the players take their actions. In a dynamic game, the players formulate strategies to act successively, while in a static game, the players formulate, decide their strategy, and act simultaneously [45].

Bipartite graph

A bipartite graph or bi-graph in the mathematical field of graph theory is a set of graph vertices, i.e., it points to where multiple lines meet, decomposed into two disjoint sets. This means they don't have any common element and no two vertices within the same graph are adjacent.

In vehicular edge computing, bipartite graph appears naturally for solving the NP-hard energy optimization problem. The limit of performance gain due to the resource offloading methods where all the vehicles offload their tasks to the same VEC (Vehicular Edge Computing) server. To address

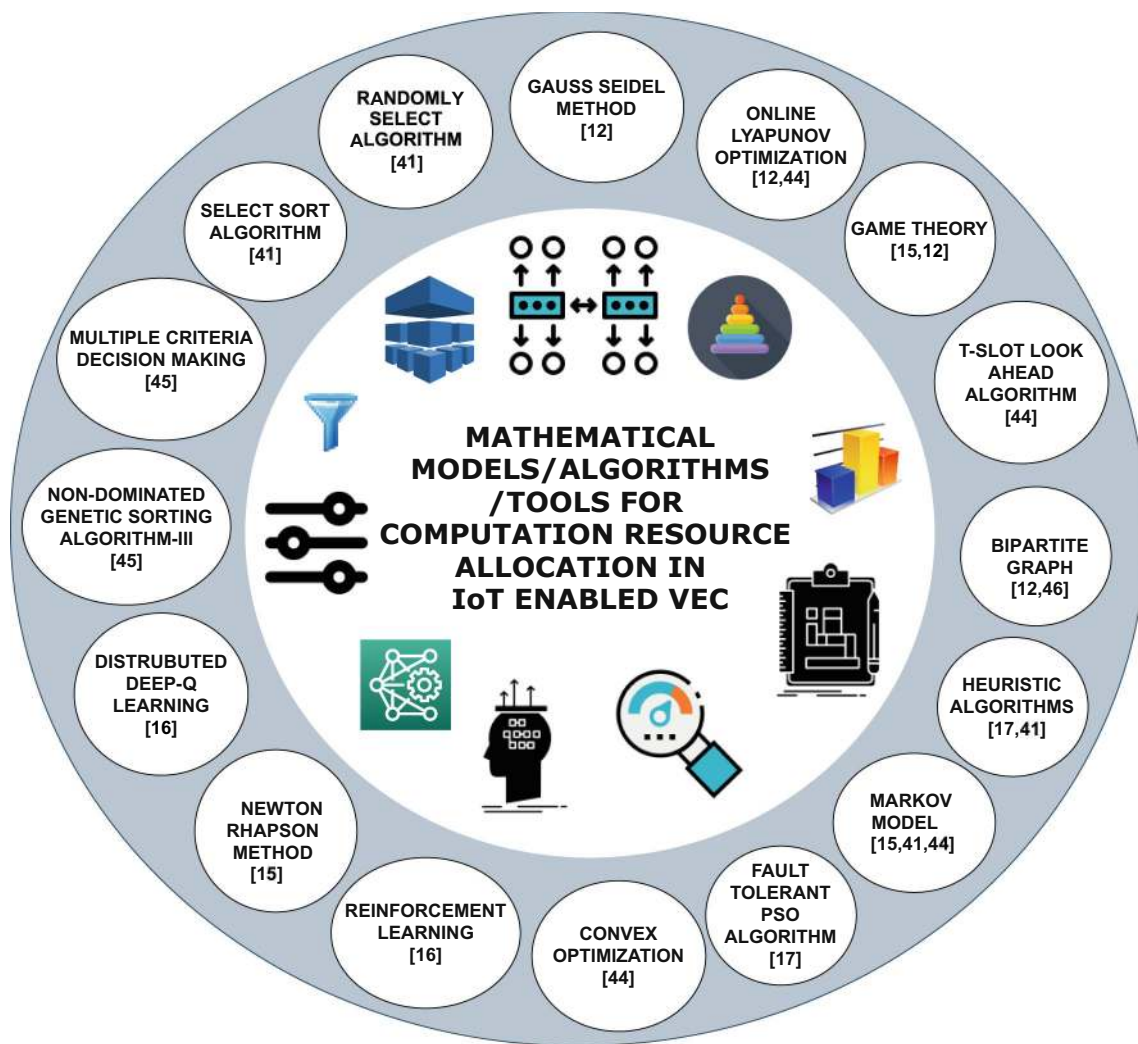


Fig. 4 Figure showing various mathematical models along with the references of the research studies in which they are used for computation resource allocation in VEC

the problem mentioned above, the authors in [40] proposed the resource allocation scheme for a multi-user multi-server VEC system. Firstly, the joint load balancing problem is formulated as a mixed-integer non-linear programming problem to maximize system utility. Notably, the authors have considered IEEE 802.11p protocol for modeling the system utility. So we split the problem into two sub-problems and develop a low complexity algorithm to make VEC server selection and optimize computation resource jointly. The problems are non-linear, which can be solved by the rounding method by constructing a bipartite graph. The rounding technique consists of two steps as follows: (i) Construction of a weighted bipartite graph to create the relationship between vehicles and RSUs (road-side units). (ii) Finding a maximum match to get the integer solution based on the bipartite graph.

V2X networks enabled with 5G are a promising technology when it comes to VEC. To solve the problem of end-to-end delay in transmission and backhauling in net-

works, the authors in [29] have formulated a URLLC resource allocation strategy and joint computation strategy problem. Then, a Bipartite graph is used to decide the channel allocation and minimize the interference in the network to increase the reliability in URLLC V2X communication. Each vehicle in the system is considered as a vertex in the bipartite graph. An overview of the model presented in [29] is given in Fig. 5.

Markov model

Markov model is a stochastic model used for modeling dynamic systems. There are four common Markov models used in different criteria. The four Markov models are Markov chain, Markov decision process, semi-Markov process, and hidden Markov model [46]. In [35], the authors have used Markov decision process to formulate the problem of resource allocation. This model is defined by the reward func-

tion, state-space, and action space. After this, the authors have solved the problem by adopting deep reinforcement learning technology. In [37], to govern the edge cloud servers, the authors have used the Markov decision process that collaborates with the SS (state search) algorithm to get the location of the vehicle. And in [26], the authors have proposed a suitable offloading scheme that considers both computational limitations and delay of the vehicular edge network. This problem is further formulated by using the semi-Markov process and later solved with DRL technology.

Heuristic algorithm

Heuristic Algorithm is designed to solve problems quickly and efficiently without sacrificing much accuracy and precision. This algorithm is often used to solve the NP-hard problems. For example, in [20], the authors have used a Heuristic algorithm to solve the formulated NP-hard optimization problem, which is non-convex in nature. This algorithm was also designed for reliability maximization. The results also show that the heuristic algorithm's reliability performance is superior to that of the Genetic algorithm and Sorting algorithm as it jointly considers communication and computational competence along with the non-success rates.

Vehicular fog computing (VFC) has become an important model which unloads the computational tasks from the cloud to the edges of the network. Due to this, the computational capabilities in vehicular applications have improved [4]. However, it is still a challenge because of the limited resources in fog computing. In [47], the authors have formulated the problem of allocating resources to fog computing in vehicular applications. Then the authors have proposed a heuristic algorithm to find the optimal solution for the problem formulated effectively. Then it is combined with reinforcement learning (RL) algorithm. But at first heuristic algorithm provides short term solution for the formulated problem, and then it is combined with the RL algorithm for long-term solutions, which makes the resource allocation more efficient.

Fault-tolerant particle swarm optimization (FPSO-MR)

Particle swarm optimization is an optimization algorithm used for optimizing non-convex functions. It is a type of heuristic algorithm with a lesser amount of complexity. In this paper [20] the authors have used FPSO-MR to achieve maximum reliability considering the latency constraints. Since the optimization problem discussed in the paper is a non-convex binary function which is also NP-hard, it cannot be easily converted to a convex function. In this algorithm, there can be more than one solutions. This algorithm works based on the movement of the particles in the search space. Each

particle has its velocity and position. The optimal solution is obtained when the particle swarm has acquired the best position. This iterative process is repeated until the particle has settled down on the best position, which is the solution.

T-slot look ahead algorithm

T slot look ahead algorithm is designed to evaluate the performance of scheduling algorithms with an analytical model approach.

In [36], the goal is to minimize the long-term computing quality loss by specifying the resource required and the quality of each running task expected. A mix-integer non-linear stochastic optimization problem is formulated to optimize the allocation of computing resources and task placement. The unpredictable network states and the high computational complexity of the formulated problem are responsible for the long-term optimization issue. It is first decomposed into a series of one-slot problems, and then an iterative algorithm called as T slot look ahead algorithm is proposed to get a computationally efficient solution.

DC (difference of convex) programming technique

When the mathematical programming problems deal with functions and those functions can be represented as the difference of two convex functions, they are called DC programming problems. For example, in [36], authors have proposed an ARAEUE algorithm. This algorithm consists of multiple integer non-convex functions. DC programming technique is used to solve this issue where acceptable convex approximations replaced the non-convex functions and restrictions, and iterative solutions were provided.

Reinforcement learning algorithm

Reinforcement learning is an algorithm in which intelligent agents take suitable actions in an environment to maximize the reward. In [28], this algorithm is used to optimize the computational offloading scheduling problem formulated to minimize the energy consumption and delay in the long term. This algorithm is further used to find suitable values in computational offloading schemes for multiple cloud computing in vehicular edge networks.

Newton–Raphson method

In [25], the authors have formed an optimization problem and named it as P1.P1 problem consists of multiple variables which can be solved by convex optimization algorithm like dual decomposition algorithm. This algorithm will create a considerable overhead. To reduce the complexity and overhead, some variables in the P1 are removed using approx-

Table 4 Classification of frameworks according to their underlying technologies

Underlying technology	References
SDN	[20,34]
Blockchain	[7,37]
AI	[7,26,28,35,47]
Cloud computing	[28,30,33,34,37,39,58]

imation, and then the optimization problem comprises only one variable, which can be optimized using the Newton–Raphson convex optimization algorithm.

Distributed deep-Q learning

Distributed deep-Q learning is a type of reinforcement learning algorithm which uses neural networks for maximizing optimization. In [28], this algorithm uses its deep neural networks to minimize overall computational time and energy consumption for processing the computational task of the i th vehicle at j th server in computational offloading schemes for multiple cloud computing in vehicular edge networks.

Non-dominated sorting genetic algorithm-III (NSG-III)

Multi-objective optimization plays a crucial role in science and technology, especially in vehicular applications where multiple objective functions are involved, and optimal solutions are proposed obviously with some trade-off between the various objectives. There are no unique solutions for multi-objective optimization or pareto optimization problem which affects multiple objective functions. Those solutions are called as non-dominated or Pareto optimal, which are found after some iterations. With the emergence of IoV, the computational capacity of vehicular networks has significantly improved. And to further enhance this, the authors in [39] have proposed a computational offloading scheme called V2X-COM (vehicle to everything communication). This scheme uses V2X (vehicle to everything) technology to transmit data in vehicular edge computing applications. Then, to provide an effective offloading plan, the non-dominated sorting genetic algorithm-III (NSGA-III) is proposed. This algorithm is adopted for improving the usage of edge server devices and reducing latency.

Multiple criteria decision making (MCDM)

Multiple criteria decision making (MCDM) analysis was first popularized in 1979 by Stanley Zionts. After that, this analysis and its mathematical modeling solved various issues in different fields around the world. This model is mainly

concerned with providing different optimal solutions and different plans rather than unique solutions and plans for solving the problems involving multiple criteria. In [39], after the NSGA-III is adopted, which provides the most optimal solution globally for the computation offloading strategies, the most optimal offloading plan is selected by multiple criteria decision making (MCDM) in the final iteration.

Simple additive weighing (SAW)

Simple Additive weighing (SAW) was first proposed by Fishburn. It is one of the simplest and oldest MCDM methods for providing multiple alternative solutions when solving multiple criteria. In [39], after the NSGA-III is adopted which provides the most optimal solution globally for the computation offloading strategies, the most optimal offloading plan is selected by simple additive weighing (SAW) along with the help of MCDM in the final iteration.

Bracketting

In [33], this is the phase-1 algorithm to calculate the minimum amount of edge resources in the particular time slot. In this algorithm, some initial value of $C_n(t-1)$ is fed, and the blocking probability is calculated. If the probability is higher than the threshold value of QoS, then the value of $C_n(t-1)$ is doubled. This process is repeated until the QoS requirement is satisfied.

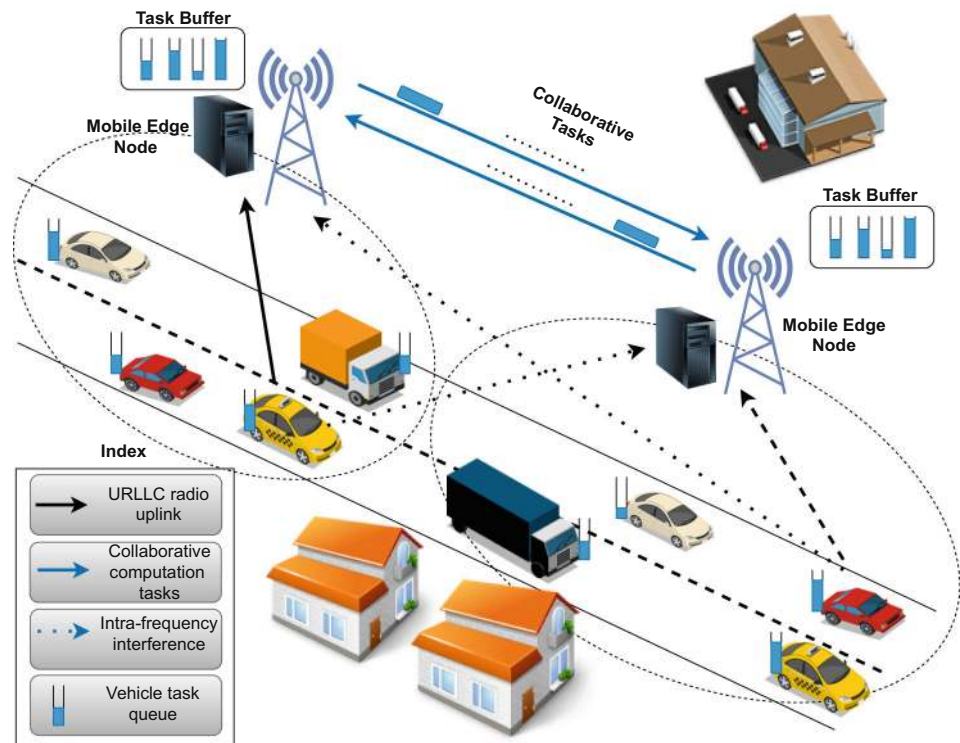
Binary searching

In [33], to find the minimum amount of edge resources and optimum resource volume for each edge cloud at a particular time slot, the authors have used a two-phase optimization algorithm. This is the phase two algorithm after the bracketting. In this algorithm, the C_{min} (edge resource) is searched, which satisfies the requirement of QoS.

Distance based heuristic algorithm (Dbha)/Euclidean distance-based algorithm

Heuristic algorithm is designed to solve problems quickly and efficiently without sacrificing much accuracy and precision. Distance-based heuristic algorithm (Dbha) is one such algorithm that takes the shortest distance as its account while responding to the requests of resources. It is observed that as the no of vehicles increasing, it is posing many challenges in limited computational capabilities. To solve this issue, many resource allocation strategies are proposed. In [34], a vehicular network with RSU (roadside unit) is proposed. Then, an effective resource allocation scheme was proposed from RSU's perspective for utility optimization. Here, Dbha is used to improve the efficiency of resource allocation. This

Fig. 5 Overview of the joint computation offloading and URLLC resource allocation for Collaborative MEC assisted C-V2X networks (Adapted from [20])



paper also shows that Dbha has great advantages over GA (genetic algorithm) and PSO (particle swarm optimization) in terms of both performance and response latency.

Maximum value density based heuristic allocation (MVDHA)

To solve the issue of excessive power and energy consumption with the massive growth of vehicular applications, the authors have proposed to expand VEC (vehicular edge computing) to VECC (vehicular edge cloud computing) in [30]. To compute the energy utilization and cost potency, a resource allocation algorithm, namely maximum value density-based heuristic allocation (MVDHA) based on maximum density value available, was proposed. Moreover, this paper shows that when the local computation energy computation is half offloading energy consumption, a maximum density value can be obtained. Based on this, MVDHA algorithm was proposed to solve the energy enhancement problem.

Randomly select algorithm (RSA)

RSA is a stochastic algorithm that uses randomized parameters based on the application to select what to do next. This randomized selection can reduce complexity. For example, a randomized sorting algorithm can choose any random pivot number for the sorting array. In [30], this algorithm uses

random resources for allocation, this random selection is based on the game theory. The energy conservation percentage using this algorithm was found to be around 53.7%. The time conservation performance of the RSA algorithm has an average of 48.7%. The performance is not consistent as it varies from 17.3 to 85.8%.

Select sort algorithm (SSA)

SSA algorithm based upon strict sorting algorithms and based on this sorting it allocates the required minimum resources. In terms of energy conservation, it provides an average saving of 59.9%. The performance of SSA in terms of time conservation has an average of 87.8%, which is better than the result obtained in RSA and MVDHA.

Summary of this section

In this section, various schemes to optimize and resolve related RA frameworks in VEC are being discussed. Different problems have been discussed along with their resolution, like using the Markov decision process to formulate the resource allocation problem for the task with different types and further enhancing it with deep reinforcement learning, Lyapunov, and DC programming based resource allocation improving user experience (ARAEUE). ARAEUE is built on the Lyapunov optimization technique, which decomposes a given problem into several one-slot mix-integer non-convex

sub-problems. Again, a CCORAO scheme separates the given optimization problem into two sub-problems with the help of a distributed computation offloading and resource allocation (DCORA) algorithm. The issue of allocating edge computation resources in vehicle mobility is solved using a lightweight 2-phase algorithm based on bracketing and binary search. Distance-based heuristic algorithm for utility optimization (DbHA) is suitable for time-sensitive scenarios where RSUs are involved. Then It is demonstrated that the MVDHA algorithm achieves an optimal balance of the number of vehicles on each server.

A summary of the algorithms used by various studies for designing resource allocation in vehicular edge computing scenario has been presented in Table 3 and also presented in Fig. 4.

Classification of RA frameworks in VEC based on underlying technologies

SDN

Some of the IoV applications are computationally very intensive, and latency-sensitive [48,49]. To support these computationally intensive and latency-sensitive IoV applications, the authors of [20] have proposed to integrate fixed edge computing nodes (i.e., fixed road infrastructure) and mobile vehicles (i.e., edge computing nodes (EC nodes)). This makes the EC nodes complex and heterogeneous. SDN is used as an agent to exploit these heterogeneous nodes, and thus EC aided IoV architecture is designed. This EC-SDIoV architecture makes it easier for mobile vehicles and fixed roadside infrastructure to behave as EC nodes to provide services via SDN with minimal latency. SDN provides comprehensive network control. It helps in deploying the network and agile management of the network without interfering with other networks. It accompanies the VEC and ITS (intelligent transport system) by dealing with complex and heterogeneous computations, a massive number of vehicles, large data flow, and frequent topology changes. This type of networking provides a strong mechanism for resource utilization by providing flexibility, programmability, and knowledge of the network [34].

Blockchain

IoV applications are complex because they require some advanced techniques to execute due to vehicles' inconsistent and unpredicted behavior. Blockchain brings advantages to vehicular applications and its infrastructure management by providing a decentralized network for smart and secure IoV. Blockchain has found immense applications in IoT related areas like UAVs [50], smart grids [51], energy management

[52], 5G [53], V2G networks [54] and industry 4.0 and industrial IoT [55]. On the industry level, many leading auto-manufacturers have considered Blockchain to improve the driving application by exchanging data in V2X. For instance, Volkswagen is building a Blockchain-based tracking system to avoid the manipulation of odometer done by the sellers to increase the price of their cars [7]. With the introduction of edge computing in intelligent driving, the vehicle movement, the time-sensitivity at which data is processed, and the allocation of the resource of the EC server have become an essential factor in intelligent driving [56,57]. To remove dependencies on third-party platforms, the authors of [37] have resource transaction architecture that is based on Blockchain. This increases system scalability, and it also allows the service providers to join/exit at any time. The Blockchain is a technique based on cryptography that allows direct trading among participants. The decentralized feature of blockchain can help in achieving the normalization of data and connect centralized databases. This decentralized feature is the actual reason why Blockchain eliminates the interference of third-party platforms.

Artificial intelligence

The new era of modern vehicles has many complex applications that require intelligent decision-making. Hence, the advancement of artificial intelligence (AI) derived its path into IoV. With the emergence of modern vehicular applications, vehicular edge computing (VEC) can solve the issue of complex computation. VEC is composed of many computing devices which are on the roadsides or in vehicles. Such architecture helps for communication, data sharing, and computation. AI is being applied in solving many challenges in vehicular applications. With the advancement of AI in IoV applications, there has been a revolutionary change in excelling self-driving, object detection, trajectory building, real-time navigation, video surveillance, traffic control, etc. Various data is being fed to an AI model, which can be a combination of game-theoretical decisions, logical reasoning, and multi-agent Reinforcement Learning (RL). With these data, the vehicles can make various decisions [7].

In [26], the authors model their resource allocation problem using a Markov decision process and use Q-learning-based reinforcement learning to arrive at the solution. They avoid dimensionality using a deep reinforcement learning (DRL) method to find the action-value function for the Q-learning. The authors of [35] have followed a similar approach to develop a scheme that is adaptable to changing MEC environments. The authors reported that although deep reinforcement learning has its advantages in solving resource allocation problems, its performance is however highly dependent on the training dataset. Hence, other training datasets have to be used for different regions of operation,

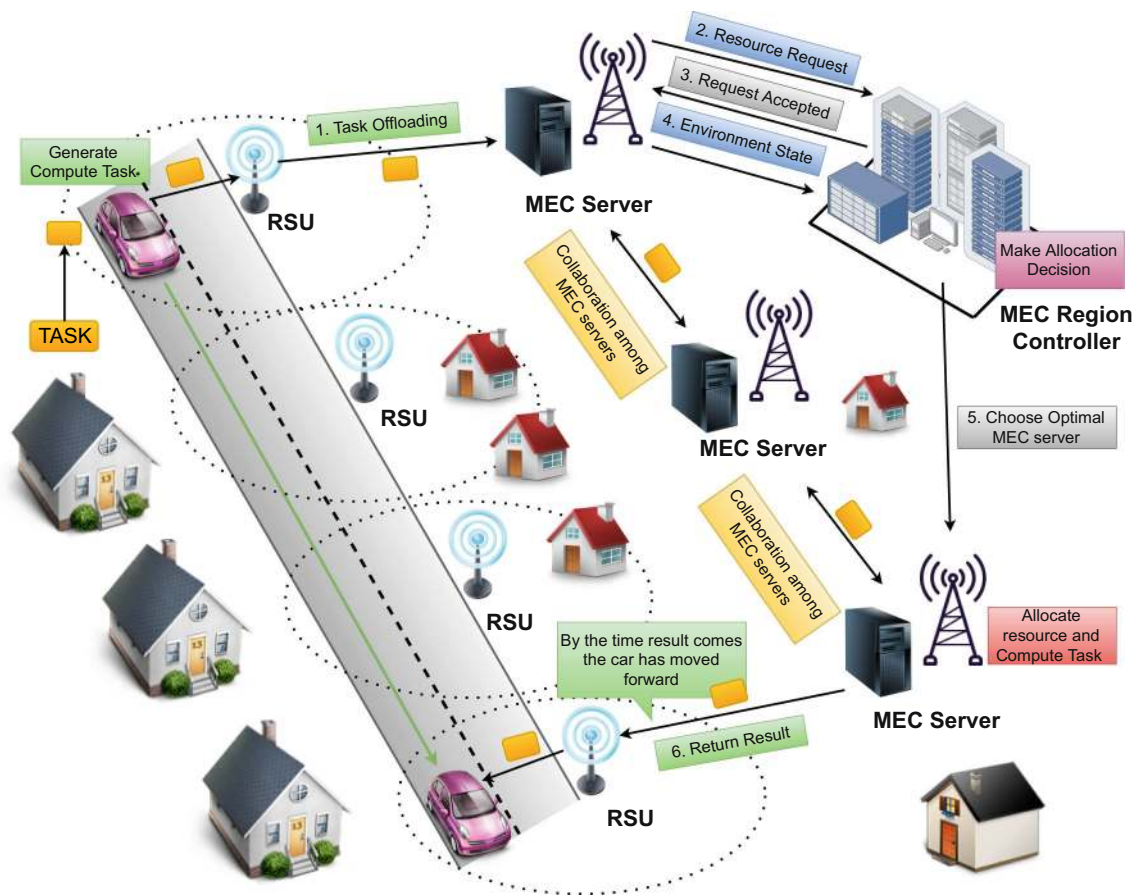


Fig. 6 Overview of the regional intelligent resource allocation in MEC based vehicular network . (Adapted from [35])

i.e., the models would have to be retrained. This is an open research issue in using deep learning techniques for resource allocation. An overview of the framework proposed by [35] is given in Fig. 6. The authors of [28] have also used DRL and Q-Learning techniques to find the best offloading decision. They propose to employ several deep neural networks in parallel to achieve quick convergence. Their simulations show that their proposed scheme performs much better as compared to other benchmark schemes.

Cloud computing

Cloud computing has been one of the critical technologies in the IoV. It provides high-performance computation, storage, and networking services. In recent years there has been a lot of studies on load balancing strategies of cloud computing. In [58], the authors use a system model in which computation offloading can be done to both MEC servers or cloud servers depending on the characteristics of the task to be offloaded. The computational resources of the MEC are lesser compared to that of the cloud server. Offloading by the vehicles to MEC/cloud servers is done through the RSU. The authors of [28,59,60] also use a similar system model in

their work. In [33], authors consider a model which uses neither edge devices nor cloud servers but edge-cloud devices in their system model. These edge-cloud devices could be stationed either at the cellular (LTE/5G) towers or at SDN/NFV enabled data centers installed by telecom equipment manufacturers (such as Ericsson) or platform vendors such as AWS. In [34], the authors employ the concept of vehicular clouds. A vehicular cloud is a temporary and localized cloud that is formed by several close-by vehicles in a VANET. The Vehicular cloud can not only increase the connectivity range of an RSU, but it can also serve offloading requests by itself either through individual vehicles or in a cooperative manner.

Summary of this section

Technologies like SDN exploits modified EC nodes which are complex and heterogeneous. SDN is used as an agent. Thus, EC aided IoV architecture is designed, making it easier for mobile vehicles and fixed roadside infrastructure to behave as EC nodes to provide services via SDN with minimal latency. To remove dependencies on third-party platforms, a transaction architecture is designed which is based on Blockchain. The decentralized feature is the actual rea-

Table 5 Summary of works surveyed

[References]	Problem addressed	Formulation/modelling	Solution	Parameter/s optimized	Underlying technology
[29]	Joint optimization of power consumption and network stability in MEC assisted Cellular V2X	Non-deterministic polynomial-time hard problem, two subproblems: 1. URLLC RA and 2. Computation resource decision	Online Lyapunov optimization method to solve CO. Gauss–Siedel method to compute CPU frequency	Delay optimization and network/system utility	–
[26]	Service hole problem in dynamic networks with static edge servers in MEC scenario	Semi-Markov process with Stochastic traffic, time-varying comm. and CO requests	A novel VEC network architecture: vehicles function as mobile edge servers. Q-learning & DRL to obtain optimal CO and RA	Power/energy optimization	AI
[35]	Computationally intensive IoVs require minimal delay in MEC. Sudden increase in computations difficult to handle due to limited resources	RA modelled as Markov decision process	A regional intelligent vehicular system with dual MEC planes. MEC servers lying in same region co-operate to achieve resource sharing	Delay optimization and QoS	AI
[36]	Need for low latency and high reliability among capacity-constrained and distributed MEC nodes. Long-term satisfactory UE for dynamic and uncertain vehicular environments	A mix-integer non-linear stochastic optimization problem for computing quality loss minimization considering queuing, dynamics, throughput constraints, and worst-case delay bound	Adaptive RA for enhancing UE in VEC networks	Delay optimization	SDN
[37]	Need for a collaborative MEC-cloud approach for offloading services to vehicles	Jointly optimizing CO decisions and RA—A non-convex NP hard problem	Collaborative CO & RA (CCORAO) scheme and distributed CO and RA algorithm (DCORA)	Delay optimization, QoE, user satisfaction, network/system utility	Blockchain and cloud computing
[38]	Present resource-constrained vehicles do not meet low/ultra-low latency demands in delay-sensitive applications such as autonomous driving	Offloading modelled as a constrained delay optimization problem	Fiber-wireless (FiWi) technology that provides centralized network management and supports multiple comm. techniques	Delay optimization	SDN, cloud computing

Table 5 continued

[References]	Problem addressed	Formulation/modelling	Solution	Parameter/s optimized	Underlying technology
[25]	Need for ubiquitous connections and high QoS for vehicles in a vehicular network	Vehicles divided into groups and RA formulated as convex optimization problem	A mobility-aware task offloading along with a location-based offloading scheme	Delay optimization	Cloud computing
[28]	Limited computational and battery capacity makes long term network participation difficult for vehicles in vehicular edge-cloud computing networks	An integration model of CO and RA optimizing weighted sum of energy consumption and latency as a binary optimization problem	A distributed DL approach to find near-optimal CO decisions using a set of DNNs parallelly	Power/energy optimization	AI and cloud computing
[39]	Vehicles have computationally intensive tasks but limited storage and computation resources.	Resource utilization and Latency reduction in IoV modelled as a multi-objective optimization goal	A CO method that employs V2X with edge computing. A non-dominated sorting algorithm-III (NSG-III) for balanced offloading using additive weighting and multiple criteria decision making	Delay optimization and network/system utility	Cloud computing
[33]	The difficulty to enhance precision in resource provisioning among distributed edge clouds in high mobility scenarios to optimize QoS and provisioning costs	An optimization problem to minimize cost of provisioning resources at edge cloud with a specified service blocking probability threshold	A stochastic traffic analysis based framework to optimize resource provisioning cost and keep blocking probability within a preset limit	QoS and network/system utility	Cloud computing
[34]	Intelligent RA strategies required for intensive and low-latency vehicular applications in dense networks	Modelled as a special knapsack problem—NP-hard category in time taken to obtain solution	An RSU backed network with three layers-Cloud, RSU-Cloulet and vehicular cloud for real-time response to resource request	QoS, QoE, user satisfaction	SDN, cloud computing
[30]	Excessive computing and energy requirements in heavily time and resource constrained vehicular scenarios	A block level minimum assignable resource for vehicular CO	A three-layer VECC framework to support real-time computation augmentation, energy conservation, and interconnection CAVs at network edges	Power/energy optimization	Cloud computing

Table 5 continued

[References]	Problem addressed	Formulation/modelling	Solution	Parameter/s optimized	Underlying technology
[40]	Difficulty for vehicles to provide required computation and performance levels in computationally intensive and delay sensitive applications	Model the joint load balancing and offloading under the permissible latency constraint to maximize utility of the system	Jointly optimizing VEC server offloading and selection in a distributed way with lower overhead	Network/system utility	–
[20]	Cloud computing may result in excessive delay for latency-sensitive applications like assisted driving etc	NP-hard, non-convex optimization for maximizing reliability performance of CO jointly considering offloading in parts, and task allocation and reprocessing for maximizing reliability	A low complexity heuristic algorithm to maximize reliability in latency constrained scenarios	Maximizing reliability	SDN
[37]	Vehicle dynamicity, delay-sensitive data processing and RA at EC server are challenges in intelligent driving	Objective function to maximize user satisfaction and three models to optimize user satisfaction: 1. Markov based state prediction, 2. Resource requirement, 3. Auction	Resource transaction architecture based on dynamic edge resource allocation and blockchain using double auction technique to optimize user and service provider satisfaction	Delay optimization, QoE and user satisfaction and network/system utility	blockchain and cloud computing
[47]	Limited number of vehicles are able to use fog computing due to its resource restrictions	Allocating limited fog resources to vehicular applications such that the service latency is minimized and user satisfaction is improved, by utilizing parked vehicles	A VFC RA algorithm that considers both short-term and long-term RAs obtained from a heuristic and RL (reinforced learning) algorithms respectively	Qos	AI

CO computation offloading, RA resource allocation, QoS quality of service, QoE quality of experience, UE user experience, AI artificial intelligence, SDN software defined network

son why Blockchain eliminates the interference of third-party platforms. Also, using DRL and Q-learning techniques, we can find the best offloading decision where it is proposed to employ a number of deep neural networks in parallel to achieve quick convergence. Lastly, edge-cloud devices can be used in the system model along with the Vehicular cloud, which not only increases the connectivity range of an RSU but can also serve offloading requests by itself either through individual vehicles or in a cooperative manner. Table

4 presents a classification of RA frameworks in VEC based on underlying technologies.

Current issues and future research directions

Based on our survey, we find that there is a lot of scope for research in resource allocation in vehicular edge computing (VEC) environment. Some of the current issues and future research directions in this area are as follows.

- 1 **Prototype testing of proposed frameworks/algorithms:** Almost all the frameworks proposed in this area have been only verified with the help of simulations. Nevertheless, simulation platforms cannot account for various actual life events that take place around a vehicle on the road. Hence, a more robust solution would require prototype testing before it can be considered for market deployment.
- 2 **Issues in AI-enabled offloading:** Most of the offloading schemes greatly rely on the availability of a dataset that covers all practical scenarios like different traffic conditions in urban and rural scenarios. Many of the proposed resource allocation schemes use AI and DRL-based methods to solve their formulated problems. Although methods like DRL have advantages in solving such problems, it is very heavily dependent on the training data. If the data changes, then the rules based on which the transition of states takes place also change, and subsequently, the network needs to be trained again. So future works will need to address this issue. Also, it is to be ensured that the data sets used for the training include a sufficient amount of data with all necessary features [35].
- 3 **Privacy leakage:** Intelligent transportation systems and vehicles collect many user data to process over a long period of time, and these are transmitted through wireless medium [61,62]. Thus, adversaries can try to compromise the data and break user privacy. Significantly when offloading tasks and communicating with edge servers, the network needs to be protected against spoofing and impersonation attacks [37]. Hence, future frameworks need to be robust and secure along with implementing efficient resource allocation schemes.
- 4 **Low computation and high speed algorithms for resource allocation:** Vehicles in a vehicular network face the constraint of having limited onboard power but at the same time have to do computationally intensive work for implementing resource allocation schemes. Moreover, such schemes require a lot of time, whereas vehicular networks are very dynamic and need to be very fast [29,30]. So, low power consumption and high speed are critical parameters that a resource allocation scheme should meet to be feasible for such scenarios. Future studies can explore using emerging technologies like FPGA and quantum computing to decrease the computation time and at the same time consume less power.
- 5 **Need for standardization:** At present, there does not exist any standards for VEC and RA in VEC. However, the development of these standards can enable interaction between numerous operators, users, and vendors on the platform. The European Telecommunications Standards Institute (ETSI) has, however, begun the development of multi-access edge computing (MEC) standards which can be used to provide various services to connected vehicles [63]. In addition, the 5G Automotive Association (5GAA) is also involved in developing use cases, architectural requirements, and deployment strategies for MEC in VEC [64]. At present, 5G networks are being deployed rapidly around the world, and hence 3GPP's 5G Standards are expected to influence the standardization of VEC and RA in VEC heavily.
- 6 **Role of 5G and beyond:** 5G and beyond can play a significant role in vehicular edge computing (VEC). The current works of the VEC are still in the early stages as the 5G communication infrastructure has not been standardized yet, and the fifth-gen and beyond network is still in the development stage. With the help of the new generation communication infrastructure, VEC can become fully functional, leading to the reduction of computational in resource-constrained users. The network can optimize mobile resources by allocating computational resources and processing extensive data before sending it to the cloud for computing. Due to the 5G and beyond infrastructure, the latency and response times could be improved, and autonomous driving could become a reality. This infrastructure will not only support computational resource allocation but also can be used for content delivery in vehicle infotainment systems. Since the latency is improved, it can be used in all latency-critical applications like autonomous driving. The computational offloading process can become more efficient and reliable. The ability of VEC to interact with 5G and beyond networks in traffic routing and resource allocation can lead to application portability which can save tons of works of developers for designing multiple versions of VEC in vehicular edge computing. Due to the 5G and beyond communication, the energy efficiency of the mobile edge node can be improved drastically, which will prolong the battery span of the nodes, and the need for frequent replacing of the battery will not be required. With the rapid evolution of wireless communication and networks, it is believed that artificial intelligence and machine learning, in particular, will play vital roles beyond 5G and 6G. The machine learning algorithms can be deployed at the VEC nodes for predicting the appropriate communication bandwidth. Artificial intelligence can significantly improve the perception of vehicular edge networks. It can assist in the optimal allocation of resources for different issues in vehicular applications. AI applies deep learning and machine learning for various resource allocation problems in VEC. The computation offloading and resource allocations are NP-hard and non-convex problems, and the variables involved in it vary with respect to time and environment. The solutions to this kind of problem can be done quickly using deep reinforcement learning (DRL). The

DRL algorithms are complex, but the problems could be solved on the cloud due to reduced latency in 5G and beyond networks. The result can be immediately transmitted to the vehicular node with a minimum amount of delay. This is all possible due to the high bandwidth of 5g and beyond infrastructure. These time-constrained problems can be computed on a cloud, and the result could be directly transmitted to the vehicles with minimum delay [65].

- 7 Scalability in IoV: There is a rapid increase in the number of vehicles around the world. This requires more efficient scheduling of resources for low latency, more reliability, and rich computational and storage capability from time to time. However, VEC is very much unevenly distributed in vehicular networks. The density of vehicles varies with time in different areas. Therefore, adaptability is a crucial issue for varying network conditions [5].
- 8 Artificial intelligence (AI): AI needs a malignant-free environment for good performance. Otherwise, it makes the VEC vulnerable to cyber-attacks which may threaten the road-safety. These attacks disrupt the algorithms and system's behavior resulting in wrong decisions that may endanger people's lives. To solve this issue, AI and Blockchain can be integrated.
- 9 Blockchain: With the introduction of blockchain, the computational complexity and latency increase which may harm the performance of vehicular applications [7]. Thus, there is a need for high-level multi-layered VEC architecture to meet the requirements of vehicular applications.
- 10 Cloud computing: In vehicular cloud computing, the mobility of the nodes is very high, and there is a continuous change in the topology. That is why it is challenging to secure vehicular communication, authentication, and the interoperability of different clouds. By implementing cache services, cloud computing can enhance the efficiency of vehicular applications. Further research is a need in this field for improving the resource allocation schemes for better vehicular communications.
- 11 SDN: SDN also has its disadvantages, especially when it comes to security [66]. It also increases the complexity of computational resources. Further research and more paradigms are needed to make SDN safer.

Conclusion

In recent years, we can see a massive growth in intelligent vehicles, which has led to rapid evolution in VEC architecture. However, due to the vast amount of data generated by these vehicles, the computation of these data on a constrained system is a significant challenge. This survey discussed the computational resource allocation for VEC

architecture using technologies like SDN, Blockchain, AI, and Cloud Computing. We have further discussed the mathematical models used by these technologies and parameters optimized while allocating computational resources. Finally, we have highlighted some of the current significant road-blocks in implementing VEC on a massive scale and its integration with upcoming technologies like 5G and beyond. We have also highlighted the future research directions in VEC.

Acknowledgements This work was supported by DST-SERB (Science and Engineering Research Board (SERB)) funding under Project Grant File No. ECR/2018/001479 received by Dr. Vinay Chamola.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (iot): a vision, architectural elements, and future directions. *Future Gener Comput Syst* 29(7):1645–1660
2. Jeschke S, Brecher C, Song H, Rawat DB (2017) Industrial internet of things cybermanufacturing systems, 1st edn [Online]. <http://lib.ugent.be/catalog/ebk01:371000000902951>
3. Contreras-Castillo J., Zeadally S., Guerrero-Ibanez J. A. (2018) Internet of vehicles: architecture, protocols, and security. *IEEE Internet Things J* 5(5):3701–3709
4. Chalapathi G, Chamola V, Vaish A, Buyya R (2021) Industrial internet of things (iiot) applications of edge and fog computing: a review and future directions. In: *Fog/edge computing for security, privacy, and applications*, pp 293–325
5. Liu L, Chen C, Pei Q, Maharjan S, Zhang Y (2020) Vehicular edge computing and networking: a survey. In: *Mobile networks and applications*, pp 1–24
6. Raza S, Wang S, Ahmed M, Anwar MR (2019) A survey on vehicular edge computing: architecture, applications, technical issues, and future directions. In: *Wireless communications and mobile computing*, vol 2019
7. Hammoud A, Sami H, Mourad A, Otrok H, Mizouni R, Bentahar J (2020) Ai, blockchain, and vehicular edge computing for smart and secure iov: challenges and directions. *IEEE Internet Things Mag* 3(2):68–73

8. Dastjerdi AV, Buyya R (2016) Fog computing: helping the internet of things realize its potential. *Computer* 49(8):112–116
9. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: MCC '12. 1. Association for Computing Machinery, New York, pp 13–16
10. Sun X, Ansari N (2016) Edgeiot: mobile edge computing for the internet of things. *IEEE Commun Mag* 54(12):22–29
11. Liu L, Chen C, Pei Q, Maharjan S, Zhang Y (2020) Vehicular edge computing and networking: a survey. In: *Mobile networks and applications*
12. Boukerche A, Soto V (2020) Computation offloading and retrieval for vehicular edge computing: algorithms, models, and classification. *ACM Comput Surv*. <https://doi.org/10.1145/3392064>
13. Zeadally S, Hunt R, Chen Y-S, Irwin A, Hassan A (2012) Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommun Syst* 50(4):217–241
14. Chen S, Hu J, Shi Y, Peng Y, Fang J, Zhao R, Zhao L (2017) Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Commun Stand Mag* 1(2):70–76
15. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. *IEEE Internet Things J* 3(5):637–646
16. Dziauddin RA, Niyato D, Luong NC, Izhar MAM, Hadhari M, Daud S (2019) Computation offloading and content caching delivery in vehicular edge computing: a survey. [arXiv:1912.07803](https://arxiv.org/abs/1912.07803)
17. Yuan S, Fan Y, Cai Y (2019) A survey on computation offloading for vehicular edge computing. In: *Proceedings of the 2019 7th international conference on information technology: IoT and smart city*, pp 107–112
18. Chamola V, Sancheti A, Chakravarty S, Kumar N, Guizani M (2020) An iot and edge computing based framework for charge scheduling and ev selection in v2g systems. *IEEE Trans Veh Technol* 69(10):10569–10580
19. Li B, Peng Z, Hou P, He M, Anisetti M, Jeon G (2019) Reliability and capability based computation offloading strategy for vehicular ad hoc clouds. *J Cloud Comput* 8(1):1–14
20. Hou X, Ren Z, Wang J, Cheng W, Ren Y, Chen K-C, Zhang H (2020) Reliable computation offloading for edge-computing-enabled software-defined iov. *IEEE Internet Things J* 7(8):7097–7111
21. Noor-A-Rahim M, Liu Z, Lee H, Ali GMN, Pesch D, Xiao P (2020) A survey on resource allocation in vehicular networks. In: *IEEE transactions on intelligent transportation systems*
22. Zheng T, Wan J, Zhang J, Jiang C, Jia G (2020) A survey of computation offloading in edge computing. In: *2020 international conference on computer, information and telecommunication systems (CITS)*. IEEE, pp 1–6
23. Jiang C, Cheng X, Gao H, Zhou X, Wan J (2019) Toward computation offloading in edge computing: a survey. *IEEE Access* 7:131543–131558
24. Lin Z, Liu J, Xiao J, Zi S (2020) A survey: resource allocation technology based on edge computing in iiot. In: *2020 international conference on communications, computing, cybersecurity, and informatics (CCCI)*. IEEE, pp 1–5
25. Yang C, Liu Y, Chen X, Zhong W, Xie S (2019) Efficient mobility-aware task offloading for vehicular edge computing networks. *IEEE Access* 7:26652–26664
26. Liu Y, Yu H, Xie S, Zhang Y (2019) Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. *IEEE Trans Veh Technol* 68(11):11158–11168
27. Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: the communication perspective. *IEEE Commun Surv Tutor* 19(4):2322–2358
28. Khayyat M, Elgendy IA, Muthanna A, Alshahrani AS, Alharbi S, Koucheryavy A (2020) Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks. *IEEE Access* 8:137052–137062
29. Feng L, Li W, Lin Y, Zhu L, Guo S, Zhen Z (2020) Joint computation offloading and urlrc resource allocation for collaborative mec assisted cellular-v2x networks. *IEEE Access* 8:24914–24926
30. Li X, Dang Y, Aazam M, Peng X, Chen T, Chen C (2020) Energy-efficient computation offloading in vehicular edge cloud computing. *IEEE Access* 8:37632–37644
31. Cai X, Wang H, Song H, Zhang Y, Cao Z et al (2019) An energy-efficiency-aware resource allocation strategy in multi-granularity provision for green computing. In: *2019 international conference on computing, networking and communications (ICNC)*. IEEE, pp 782–786
32. I. Recommendation (1994) E. 800: terms and definitions related to quality of service and network performance including dependability. ITU-T August, vol 1994
33. Tang G, Guo D, Wu K, Liu F, Qin Y (2020) Qos guaranteed edge cloud resource provisioning for vehicle fleets. *IEEE Trans Veh Technol* 69(6):5889–5900
34. Tang C, Zhu C, Wei X, Wu H, Li Q, Rodrigues JJ (2020) Intelligent resource allocation for utility optimization in rsu-empowered vehicular network. *IEEE Access* 8:94453–94462
35. Wang G, Xu F (2020) Regional intelligent resource allocation in mobile edge computing based vehicular network. *IEEE Access* 8:7173–7182
36. Sun X, Zhao J, Ma X, Li Q (2019) Enhancing the user experience in vehicular edge computing networks: an adaptive resource allocation approach. *IEEE Access* 7:161074–161087
37. Xiao K, Shi W, Gao Z, Yao C, Qiu X (2020) Daer: a resource pre-allocation algorithm of edge computing server by using blockchain in intelligent driving. *IEEE Internet Things J* 7(10):9291–9302
38. Zhang J, Guo H, Liu J, Zhang Y (2019) Task offloading in vehicular edge computing networks: a load-balancing solution. *IEEE Trans Veh Technol* 69(2):2092–2104
39. Xu X, Xue Y, Li X, Qi L, Wan S (2019) A computation offloading method for edge computing with vehicle-to-everything. *IEEE Access* 7:131068–131077
40. Dai Y, Xu D, Maharjan S, Zhang Y (2018) Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet Things J* 6(3):4377–4387
41. Gauss CF (1903) *Werke* (in german), 9. Göttingen: Königlichen Gesellschaft der Wissenschaften 763:764
42. Sauer T (2011) *Numerical analysis. Always learning*
43. He J, Li Y, Li H, Tong H, Yuan Z, Yang X, Huang W (2020) Application of game theory in integrated energy system systems: a review. *IEEE Access* 8:93380–93397
44. Guo W, Chang Z, Guo X, Jayakody DNK, Ristaniemi T (2020) Resource allocation for edge computing-based blockchain: a game theoretic approach. In: *2020 IEEE international conference on communications workshops (ICC Workshops)*. IEEE, pp 1–6
45. Ritzberger K (2002) *Foundations of non-cooperative game theory*. OUP Catalogue. Oxford University Press, no. 9780199247868 [Online]. <https://ideas.repec.org/b/oxp/obooks/9780199247868.html>
46. Wei W, Fan X, Song H, Fan X, Yang J (2016) Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing. *IEEE Trans Serv Comput* 11(1):78–89
47. Lee S-S, Lee S (2020) Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information. *IEEE Internet Things J* 7(10):10450–10464
48. Jindal A, Aujla GS, Kumar N, Chaudhary R, Obaidat MS, You I (2018) Sedative: Sdn-enabled deep learning architecture for network traffic control in vehicular cyber-physical systems. *IEEE Netw* 32(6):66–73
49. S. C. G., V. Chamola, C.-K. Tham, G. S., and N. Ansari, An optimal delay aware task assignment scheme for wireless sdn networked edge cloudlets, *Future Generation Computer Systems*, vol. 102,

- pp. 862–875, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167739X19301232>
50. Alladi T, Chamola V, Sahu N, Guizani M (2020) Applications of blockchain in unmanned aerial vehicles: a review. *Veh Commun* 23:100249
 51. Alladi T, Chamola V, Rodrigues JJ, Kozlov SA (2019) Blockchain in smart grids: a review on different use cases. *Sensors* 19(22):4862
 52. Miglani A, Kumar N, Chamola V, Zeadally S (2020) Blockchain for internet of energy management: review, solutions, and challenges. *Comput Commun* 151:395–418
 53. Praveen G, Chamola V, Hassija V, Kumar N (2020) Blockchain for 5g: a prelude to future telecommunication. *IEEE Netw* 34(6):106–113
 54. Hassija V, Chamola V, Garg S, Krishna DNG, Kaddoum G, Jayakody DNK (2020) A blockchain-based framework for lightweight data sharing and energy trading in v2g network. *IEEE Trans Veh Technol* 69(6):5799–5812
 55. Alladi T, Chamola V, Parizi RM, Choo K-KR (2019) Blockchain applications for industry 4.0 and industrial iot: A review. *IEEE Access* 7:176935–176951
 56. Jindal A, Aujla GS, Kumar N (2019) Survivor: a blockchain based edge-as-a-service framework for secure energy trading in sdn-enabled vehicle-to-grid environment. *Comput Netw* 153:36–48
 57. Aujla GS, Jindal A (2020) A decoupled blockchain approach for edge-envisioned iot-based healthcare monitoring. *IEEE J Sel Areas Commun*
 58. Zhao J, Li Q, Gong Y, Zhang K (2019) Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. *IEEE Trans Veh Technol* 68(8):7944–7956
 59. Saleem A, Khan A, Malik SUR, Pervaiz H, Malik H, Alam M, Jindal A (2019) Fesda: fog-enabled secure data aggregation in smart grid iot network. *IEEE Internet Things J* 7(7):6132–6142
 60. Kumar N, Dhand T, Jindal A, Aujla GS, Cao H, Yang L (2020) An edge-fog computing framework for cloud of things in vehicle to grid environment. In: 2020 IEEE 21st international symposium on “a world of wireless, mobile and multimedia networks” (WoWMoM). IEEE, pp 354–359
 61. I. Ullah, M. A. Shah, A. Khan, and G. Jeon, Privacy-preserving multilevel obfuscation scheme for vehicular network, *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, p. e4204, 2021
 62. Garg S, Kaur K, Kaddoum G, Ahmed SH, Jayakody DNK (2019) Sdn-based secure and privacy-preserving scheme for vehicular networks: a 5g perspective. *IEEE Trans Veh Technol* 68(9):8421–8434
 63. ETSI—Multi-access Edge Computing—Standards for MEC [Online]. <https://www.etsi.org/technologies/multi-access-edge-computing>
 64. Sabella D, Moustafa H, Kuure P, Kekki S, Zhou Z, Li A, Thein C, Fischer E, Vukovic I, Cardillo J et al (2017) Toward fully connected vehicles: edge computing for advanced automotive communications. White Paper
 65. Pham Q-V, Fang F, Ha VN, Piran MJ, Le M, Le LB, Hwang W-J, Ding Z (2020) A survey of multi-access edge computing in 5g and beyond: fundamentals, technology integration, and state-of-the-art. *IEEE Access* 8:116974–117017
 66. Alladi T, Kohli V, Chamola V, Yu FR (2021) Securing the internet of vehicles: a deep learning based classification framework. In: *IEEE networking letters*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.