International Journal of Grid and High Performance Computing
Volume 15 • Issue 2

Mobile Edge Computing Architecture Challenges, Applications, and Future Directions

Teja Sree B., Koneru Lakshmaiah Education Foundation, India
G. P. S. Varma, Koneru Lakshmaiah Education Foundation, India*
Hemalatha Indukurib, S.R.K.R. Engineering College, India
https://orcid.org/0000-0001-9372-7740

ABSTRACT

In the current era of technology, the utilization of tablets and smart phones plays a major role in every situation. As the numbers of mobile users increase, the quality of service (QoS) and quality of experience (QoE) are facing the greater challenges. Thus, this can significantly reduce the latency and optimize the power consumed by the tasks executed locally. Most of the previous works are focused only on quality optimization in the dynamic service layouts. However, they ignored the significant impact of accurate access network selection and perfect service placement. This article performs the detailed survey of various MEC approaches with service provision and adoption. The survey also provides the analysis of various approaches for optimizing the QoS parameters and MEC resources. In this regarding, the survey classifies the approaches based on service placement, network selection, QoS, and QoE parameters, and resources such as latency, energy, bandwidth, memory, storage, and processing.

KEYWORDS

energy consumption, fifth-generation, latency, mobile edge computing, Multimedia tasks, network selection, service placement

1. INTRODUCTION

Recent advances in the cost, performance, and energy efficiency of IoT devices; network technologies (such as 4G and 5G) (Liu, 2013) and distributed computing architectures have led to the explosive growth of the Internet and mobile connectivity, in turn leading to new distributed applications in areas such as transportation, healthcare, mining, entertainment, and security, such as automated vehicles, augmented reality, cloud robotics, smart homes and cities, video surveillance and streaming, and Internet of Things (IoT) applications (Best-Rowden, 2018) This has led to an unprecedented growth

DOI: 10.4018/IJGHPC.316837
*Corresponding Author

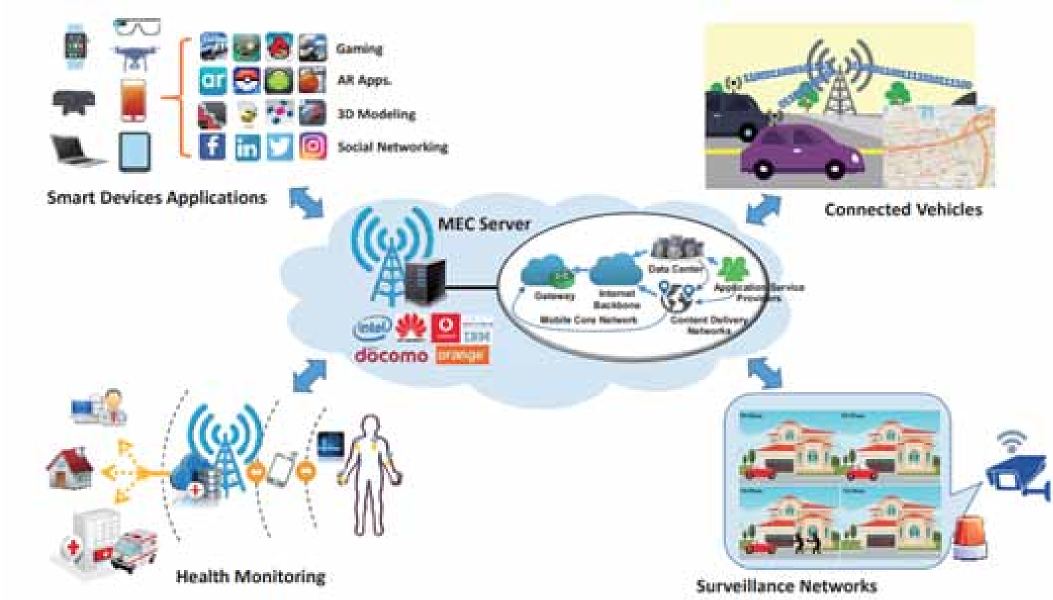
in data as well as increased the importance of latency and regulation in handling and managing data. The new distributed applications have characteristics that may be bandwidth-hungry (video surveillance, video conferencing, traffic monitoring), latency-critical (automated vehicles, robotic surgery, safety), and may cause spikes in activity at places or times (sporting events). Applications may also require high availability, low jitter, and security. Large-scale deployment of IoTs and industrial IoT devices (Sun, 2017) is expected to play a big role in the development of smart cities, which will generate large volumes of aggregated cellular data that may choke the network. On the other hand, devices such as sensors on the power grid or on oil pipelines may host latency-sensitive applications that require low latency in order to ensure that mission-critical data is transmitted and processed in a timely manner so that potential damage to people, property, and the environment can be prevented (Wang, 2018). Online video games on consoles such as Xbox Live, where reaction times are in milliseconds, have become extremely popular recently. Such games may be hosted by a distant data centre (DC), so the presence of latency and jitter could have a significant effect on gamers' experience and dramatically reduce their interest in the games. Virtual Reality (VR), Augmented Reality (AR), and other state-of-the-art human-computer interaction applications require low latency and rapid processing for complex rendering algorithms as well as large volumes of data that may need to be transferred between a user and a DC hosting the applications (Mao, 2017). There has been a rush of live streaming applications (Tran, 2017) such as SnapChat, Facebook Live, and YouTube Live, due to the proliferation of high definition (HD) video cameras on smart phones. Similarly, video surveillance applications will require high-performance computer resources to run artificial intelligence (AI) and machine learning (ML) technologies (Wang, 2018) to identify people and alert human operators in real time. These applications may generate gigabytes or even terabytes of data per second. All in all, there is a need for a compute infrastructure that can begin to address the challenges posed by these emerging applications. Today's telecom networks are not even expected to handle the enormous and rapidly varying capacity demands that will arise soon. One of the challenges of using IoT applications to their fullest potential (Taleb, 2017) is figuring out how to handle network traffic between end-users and application-hosting nodes while keeping infrastructure costs low and meeting QoS requirements of end-users, such as latency and/or throughput. One way to address the challenges presented by emerging IoT applications is to move the computations closer to end-users – that is, towards the ISP's edge network – to reduce transmission costs, decrease network latency and jitter, increase reliability, and avoid network congestion. A key idea is to create a unified ICT infrastructure using existing large-scale distributed cloud infrastructures and augmenting (Li, 2016) them with compute capacities at intermediary nodes, such as radio base stations at the ISP's edge network and inside its internal network. As shown in Figure 1, the unified infrastructure at the edge of the network, called "Mobile Edge Clouds," can host applications closer to the end users. This helps solve congestion problems and meets end users' performance expectations.

The main contribution of this article is as follows:

- In this paper, various optimization methods and its limitations are analyzed using the resource and task constraints for service selection, network selection and task scheduling process.
- A resource constraint based PSO approach is proposed for mobile task scheduling process in cloud virtual servers.

Rest of the article is organized as follows: section 2 deals with the various concepts and technologies of cloud computing. Section 3 deals with the various optimization methods. Section 4 deals with the open issues presented in MEC. Finally, section 5 represents the proposed optimization model with resource constraints along with experimental results.

Figure 1. Application sectors of MEC



2. RELATED CONCEPTS AND TECHNOLOGIES

This section presents a detailed discussion of the various types of clouds, such as MCC, FC, MEC, and cloudlets, respectively. Figure 2 presents the detailed analysis of different edge paradigms with the infrastructure implementation from CORE to EDGE (Piao, 2010). Table 1 presents the detailed comparison of MEC, FC, MCC, and Cloudlet. From the comparison, it is observed that MEC gives better QoS performance as compared to the conventional edge computing paradigms (Liu, 2015). Finally, the open research problems and challenges of MEC are effectively analyzed and used to elaborate on the new research trends, which are missed in the related work.

MCC: Mobile cloud computing (MCC) combines both mobile networks and cloud computing (CC) together. The mobile and cloud computing methods are combined and formed as the MCC model. This model provides sufficient energy resources, computation, and storage capacities to the mobile users (MUs) (Mitsis, 2019) through a centralized cloud. The fundamental architecture of MCC includes the cloud servers and MUs as presented in Figure 3.

According to the MCC, the cloud server exhibits unlimited resources, and it is far away from the MUs. Thus, the distance between the MUs and the cloud server increases the access delay. The core network of MCC has reduced computing capacity and resources. The access requests generated from the MUs are never directly executed by the core network (Chen, 2014, Yang, 2012).

Fog Computing: The MCC is widely utilised in the lower level IoT networks, but it still creates multiple problems, such as location awareness, lack of mobility support, and unreliable delays in IoT networks. The FC is used by the IoT networks to solve the above issues. The FC effectively allocates flexible services and resources to the MUs (Kumar, 2010). The FC gives support to both connected and unconnected devices. Initially, the FC was developed by Cisco Systems to provide unlimited services to the edge-connected enterprise network. FC suggests less latency as compared to conventional cloud computing approaches. In any case, the FC method suffers from the more reliant continuous wireless connection. The small interruption in the connections makes the process fail. Thus, the complex operations are not performing accurately through the FC and it supports the intelligence at LAN only (Deng, 2016). Thus, a greater number of attacks and threats are affecting

Figure 2. Simplified overview of major Edge paradigms

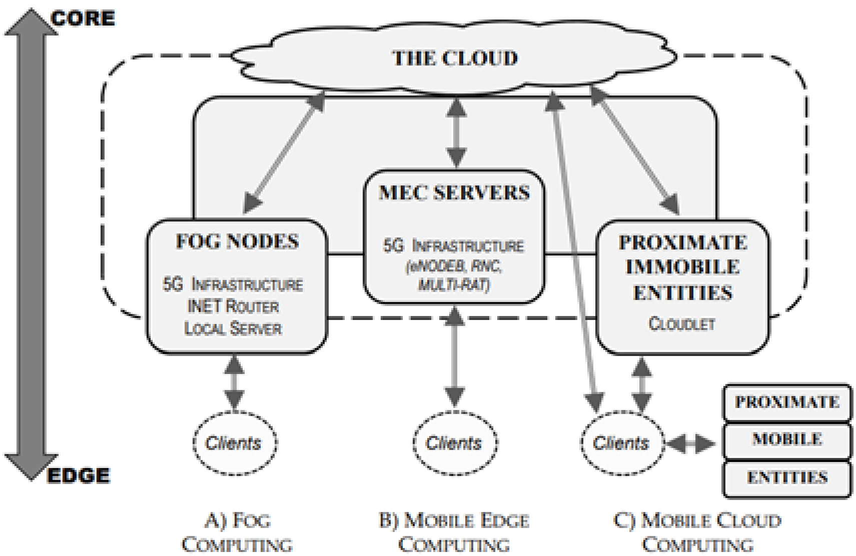


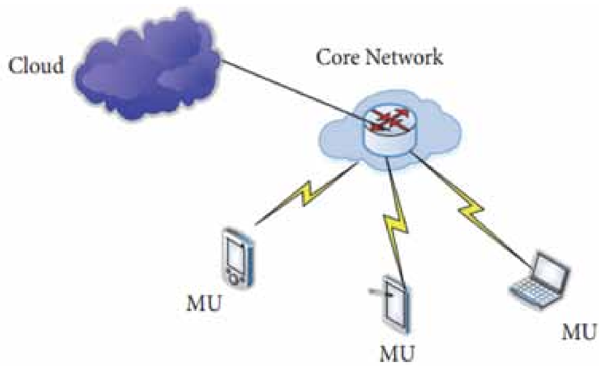
Table 1. Comparison of features of Edge paradigms

Method	Cloudlet	MCC	FC	MEC
Scalability	Average	High		
Availability	High			
Local Awareness	N/A	Yes		
Latency, Jitter	Average	Low		
Mobility	N/A	Yes		
Net. Architecture	Centralized	Distributed, Decentralized, N-Tier		
Service	Virtualization	Virtualization, Others	Virtualization	
Hardware	Servers	Servers, User devices	Heterogeneous servers	
Deployment	Network Core	Network Edge, Devices	Near-Edge, Edge	Network Edge
Ownership	Private entities	Private entities, Individuals		Telco Companies

the performance and MUs privacy information. Thus, MEC is effectively used to support security at both the IoT level and mobile-to-mobile communications. And MEC also supports the various operations of IoT networks with reduced computations.

Cloudlets: Cloudlet (Pavlos, 2018) is the small source of a cloud network formed across single MUs and groups of multiple MUs. These cloudlets are utilised mostly in office buildings, shopping centers, and public places like transportation areas and hospitals with small data servers. The main objective of cloudlet is to interconnect different cloud technologies and provide the maximum number of resources to end users with reduced latency (Chen, 2016). Cloudlets mainly depend on Wi-Fi, WMAN, and WLAN technologies. Thus, if any interruption is generated in these technologies, then single hop and multiple hop connections between MUs connected to the internet get closed. The cloudlet also suffers from privacy and security issues that are related to the access-based privacy issues (Ksentini, 2014). So, MEC is a good way to solve these problems, and it works with all kinds of communication technology standards.

Figure 3. Mobile cloud computing (MCC) architecture



The MEC: The MEC plays a prominent role in advanced 4G and 5G communications as it overcomes the problems generated by the Cloudlets, FC, and MCC systems, respectively. The MEC consists of multi-functional MCCs for performing all types of resource allocation to the end users of the radio access network (RAN) (Wang, 2015). But these methods failed to provide the maximum efficiency in the high computational environments and resulted in a reduced data offloading ratio (Srivatsa, 2013). Thus, the MEC is equipped with a consumer-oriented offloading service management unit, an operator-third party-based service management unit, and a network performance and QoE improvement service management unit as shown in figure 4. The primary objective of MEC is to route the packets between various applications based on their services. The routing is performed based on independent mode, tap mode, inline mode, and breakout mode, respectively. The secured session connection will be re-estimated between the remote server and the local MEC host during breakout mode (Nicholson, 2006). The best examples of breakout mode data transmission are enterprise LAN or local CDN applications. The secured session is maintained with the standard servers during the inline mode and transmits all the traffic to the MEC applications through the internet. The security and caching related applications are treated as inline applications that are generated in the inline mode of operation. The resource-based traffic is simulated during the tap mode and transmits all resources to MEC applications. Finally, irrespective of the data and traffic offloading ratios, the resources are transmitted over to the MEC applications during the independent mode (Li, 2020). But the application needs to register with MEC infrastructure in this mode of operation. Thus, the MEC provides the highest offloading rates with improved bandwidth and reduced network latency by performing the various modes (Yang, 2017) of operations perfectly. However, MEC technologies are suffering from various challenges in real-time scenarios. Various research has been conducted on the MEC and it is still limited. As a result, a thorough examination of various conventional MEC technologies can provide significant solutions to all of the problems encountered in the edge network. The survey also reveals the un-covered area of potential research and gives prominent directions of study.

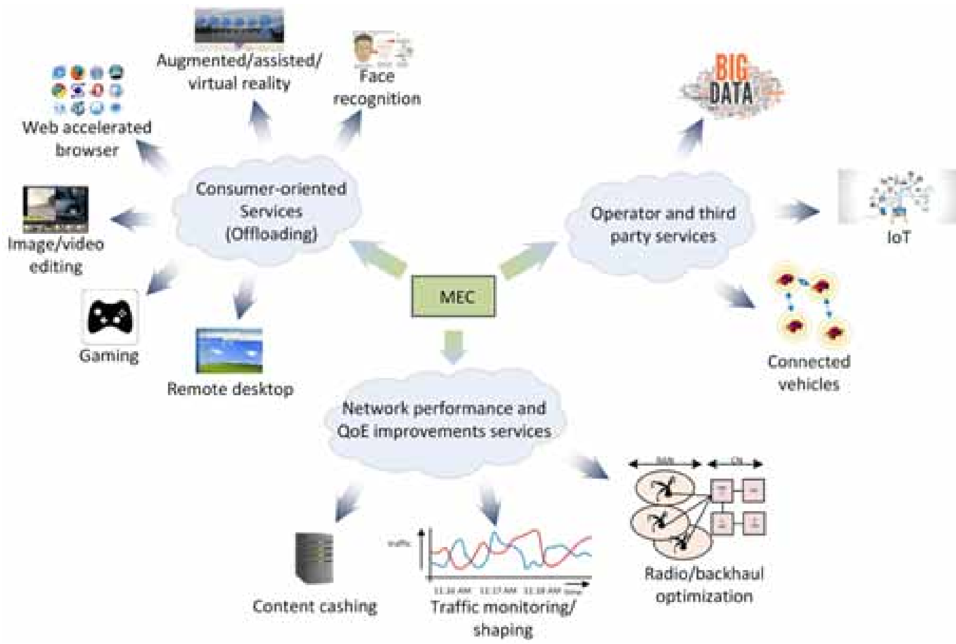
3. OPTIMIZATION APPROACHES

This section deals with the study of various optimization methods for providing efficient QoS parameters and MEC resources to the end users. These parameters provide different resources based on the MEC environment, such as end users, infrastructure, and service providers.

3.1 Service Placement Optimization Approaches

In MEC environments, service placement is the most important thing to think about, because better service placement means less latency, better network performance, and less wasted energy. In (Al-Badarneh, 2018),

Figure 4. Cloud process MEC services



researchers have focused on the main challenge of MEC, which is to increase the QoS constraints. In order to accomplish the task, the author reduced the execution time and applied a round-robin-based scheduling algorithm. The result of the approach provided a high rate of job completion within a stipulated interval of time. They have addressed the MEC issues in a cloud environment. To overcome this issue, the authors proposed an improved clonal selection algorithm, which is used to select the best resource and allocate resources to user requests with minimal time. However, this algorithm increases the make-span, response time, and cost. A discussion about selective algorithms for data offloading allocation for MEC architecture is presented in (Huang, 2018), where data offloading is allocated on demand, min-min, min-max, and heuristic algorithms are integrated into the selective algorithms. All three of the above three algorithms provide a minimised time span on allocation, and it works well only for independent jobs. Overall system performance is improved while system utilization is increased. Later, a hierarchically structured MEC is designed and implemented using knowledge management techniques (Vladyko, 2019), which are used to decrease the cost of the latency level agreement violations, improve the performance, reduce the energy consumption, and allocate the resources. The virtual machine reconfiguration is applied each time MEC and it takes more time and reduces the availability of the resources. Lakhan (2021-2022) proposed a fault tolerant workload scheduling schemes using mobile edge computing for different applications.

In (Kaur, 2018), researchers have proposed a scheduling algorithm that is most effective for MEC in Industrial IoT (IIoT). The scheduling process concentrates on the time allocation according to the job size and reduces energy consumption. They have mainly focused on QoS in terms of energy efficiency and effective MEC. The major drawback of this work is that the trade-off between energy efficiency and latency is not achieved perfectly. An improved algorithm that is dependent on software defined networking (SDN) has been introduced for maintaining the relationship between various QoS metrics of MEC has been addressed in (Peng, 2019).

Authors in (Shah, 2020) analysed the benefits of multi-access edge computing (MAEC) over conventional MEC, and then made the best use of MAEC to resolve the load balancing issue emerging in cloud computing by developing a load balancing model following virtualization resource management.

MAEC is a type of modified hybrid MEC; the execution shows highly improved convergence time and optimization results in comparison with that of the conventional MEC but failed to provide better latency across the Mobile Edge Hosts (MEH). To address this problem, in (Sechkova, 2018), the authors introduced the Network Functions Virtualization (NFV) for the MEC environment. Since hybrid clouds are being used for more and more large-scale applications, a workflow management service must be able to effectively schedule and guide the execution of the applications. This work must continue to focus on cost, make span, fairness, system-level efficiency, and other similar aspects as related algorithms.

All these parameters are covered in (Dai, 2019), where a joint load balancing approach is implemented, and where the authors primarily devised the problem of resource scheduling. After this, a resource scheduling policy is introduced to achieve task completion time, optimal resource utilization, average consumed energy, and average cost. But this SDN is suffering from Distributed Denial-of-Service (DDoS) attacks. Recently, to address these security issues in (Babou, 2020), the authors introduced the sustainable and secure load balancing scheme for MEC.

3.2 Optimization Approaches For Network Selection

The right choice of network leads to the best placement of users, a lower number of virtual paths in the network, and a better network span for the end users. In (Feng, 2020), authors designed a novel entity called dynamic network slicing involving cloud tasks and cloud service providers to perform the resource allocation process. A dynamic resource controller will perform resource control tasks by analysing the factor called traffic flow. To resolve this problem, Micro-Operator Networks (Sanguanpuak, 2018) were introduced for network slicing with resource allocation. Further The probabilistic network selection based MEC was implemented in (Forti, 2019).

In (Wang, 2019), authors proposed a QoS-aware resource control scheme in 5G networks. The utilisation of this approach increases the utilisation of the server and system throughput along with achieving per-application QoS guarantees on the selected network. In (Kourtis, 2019) a methodology to manage the workload in the MEC databases was developed with the Small Cell Architecture. The workload is split by considering the QoS of the tasks that need to be allocated to the nearest smart cells. By doing so, better handling of cloud resources can be done. Thus, the approach can improve the resource control process considering the latency optimization parameters. The major drawback of this approach is that task execution failure is not considered as an evaluation parameter, and hence resource handling can be affected. To address these issues, a seamless support learning (SSL) algorithm was introduced in (Zhang, 2020) for joint network selection with latency optimization. These mechanisms can perform scheduling with the satisfaction of users' QoS constraints. The proposed algorithms provide better results in terms of achieving fast and robust convergence.

Authors in (Alameddine, 2020) proposed a dynamic task offloading and scheduling mechanism in optimal networks using logic-based benders decomposition to provide a flexible network environment to all service providers. Energy consumption is often related to factors such as QoS parameters like power consumption levels, network loads, dynamic task timings, and so on. In (Li, 2019), the issue of inserting survivable virtual network functions (VNF) at least operational expenses is addressed. The problem of improvement is taken care of in two steps (the position of the cloud in the network and the mapping of the virtual connection) because of the limited number of assets and the need for more transmission capacity. To address all the network related issues, the (Ahn, 2019) authors introduced the Power Efficient Clustering Scheme in VNF for effective MEC performance. The mechanism initially attempts to allocate the job task to a server through a VNF. On the off chance that any of the three stages fails, the heuristic includes another adjoining server and emphasises the mapping procedure while allowing for server defragmentation, residual data transfer capacity, correspondence costs, and load adjusting. Plenty of techniques are utilised for cloud architecture prediction, but the accuracy of prediction is low. The rationale for accuracy is that requests are terribly discretionary. Models of requests differ frequently, so predicting the correct pattern is difficult. Researchers suggested doing a load forecast in MEC offers edges to some extent in comparison with the varied available services.

3.3 Optimization Approaches for Latency

Latency is the major problem in the MEC. As the number of VNFs increases in the network, the network latency also increases. In (Yala, 2018), the authors suggested an algorithm known as ultra-Reliable Low-Latency Communications (uRLLC), whose objective is to reduce the total latency of task execution when fulfilling the target limitations. Sarrigiannis, 2018) It is based on a reliable supply of resources and established the adaptive frame for cloud service. This framework renders cloud services for adaptive adjustment in accordance with conditions which occur during the user/consumer service or supply irregularities. This method mainly suffers from the dynamic allocation of resources because it results in an increase in network latency. In (Bi, 2020), the authors suggested a novel latency optimization approach for dynamic resources generated in the network. The guaranteed objectives are such that the users present in the same queue get dynamic allocations as per the proportion of their fair shares when users in diverse queues get allocations, which increases the dynamic resource usage in accordance with well-examined fairness characteristics like those in VNF. The computational complexity of this method has increased significantly. Thus, in (Yang, 2018), the authors developed a cost-effective NFVO-based MEC with low latency. The workflow scheduling is implemented based on the NFVO-based scheduling algorithm; it is used for mapping the requests of users to the suitable available resources. The workflow scheduling is generally carried out manually through low-latency mobile applications.

Further, the latency of the MEC is mainly dependent upon the VNF placement features. Thus, in (Solozabal, 2017), the authors introduced the virtual infrastructure manager (VIM), which helps in evaluating the fair behaviour of a resource-based algorithm. In this framework, two forms of sub-models, Dynamic Node Model (DNM) and Dynamic Demand Model (DDM), are presented to define the VNF placement features of resource demand as well as identify the number of computing nodes within a cloud service environment. But this method is suffering from the joint optimization of VNF placement optimization with latency mitigation. Thus, in (Leivadeas, 2019), the authors introduced the VNF placement optimization specifically for latency optimization.

Further, in (Son, 2019), the authors introduced dynamic resource provisioning for latency optimization. This technique of MEC studies the coalition creation of the virtual paths on the cloud, which results in increased latency. Thus, the dynamic change of the network increases the overlapping of virtual paths significantly, and they need to be optimised more. Thus, to optimise these dynamic network-based latency issues, the Genetic Algorithm (GA) based bio-optimization algorithm was introduced (Ruiz, 2018). The GA selects the best paths and avoids the paths overlapping. The GA is also used to reduce the network delay generated due to overflow of data in VNF. But this method failed to provide a better QoE. The drawback of this algorithmic rule is that the extent of the populace that is considered is extraordinarily small to necessitate every condition. In (Ma, 2020), the authors introduced a MEC management technique for allocating the low latency resource in accordance with the versatile needs of various kinds of QoE. The solution consists of a MEC allocation algorithm to guarantee that heterogeneous assignments are allotted suitably to prevent biased resource usage. A model-based scheme for estimating the suitable number of active QoS for operating the QoE Further, in (Alfakih, 2021), authors presented a novel QoE-based resource allocating algorithm.

Finally, the authors presented the QoS task scheduling-based latency optimization method in (Han, 2020). For assigning the task requests to the processing nodes, make use of latency-based disruption planning in fair scheduler. This method achieves a scalable and quick-response method. This method provides a solution to the latency problems by maintaining both QoE and QoS, respectively.

3.4 Optimization Approaches for Energy Resources

Energy efficiency is the major concern in the field of MEC. The energy-efficient architecture depends on the network selection, network utilization, and number of users. In (Chou, 2016), the authors introduce the service based on a demand-based network selection approach for better energy resources. Thus, in (Chen, 2019) the authors introduced a mobility-aware service composition approach which attempted to combine the resources of both networking and cloud services together. The composition of the network cloud services would be a most difficult process, which is resolved in this work by considering it as a multi-constrained optimal path problem. To optimise this problem, a multi-tenant resource allocation method (Tun, 2019) was developed with energy efficient properties. This method applied the Karush-Kuhn-Tucker (KKT) conditions to solve each sub problem that occurred in the main problem.

Further, in (Sun, 2018), the authors introduced a low-latency orchestration-based energy efficiency improvement strategy. This grouping of resources is done to satisfy the latency and efficiency demands of real-time applications. This work gathers the user requirements, which would be grouped together based on the QoS constraints. Again, this method failed to provide the QoE while satisfying the energy requirements. Thus, the service chaining concept (You, 2021) is introduced to meet these challenges. Fuzzy based security service chaining (Li, 2017) is the popular method for providing both QoS and QoE requirements with the best energy resources. Fuzzy Theory will store the standard resources in the cloud storage information and the frequently used resources in the buffer. This can speed up the process by responding to the user's requests immediately from the buffered resources. Furthermore, Edge Placement and recourse availability-based methods (Zhu, 2018) are introduced for chained MEC applications. This work designed a framework that could mitigate the impacts of cloud uncertainty and achieve good and insensitive resource allocation. The overall aim of this work is to eliminate the uncertainty problem and achieve optimal resource allocation. But these approaches are suffering from network slicing problems with reduced energy efficiency as the number of resources increases. Thus, in (Ksentini, 2020), the author introduced the network slicing based energy efficient architecture for uRLLC based 5G networks. A network-based automated service composition algorithm was proposed, which would gather the user requests along with their QoS constraints and perform the MEC process automatically by composing the required resources. However, the MEC is facing serious problems in the network slicing process with respect to standardization viewpoint. All the problems and limitations presented are presented in (Cominardi, 2020).

The solutions to all these problems are addressed by the authors in (Tomaszewski, 2020). This work developed the Distributed Autonomous Slice Management and Orchestration (DASMO) algorithm. For better emergency efficiency in each network slice, the Autonomous Slice Management and Orchestration (DASMO) algorithm was developed. After slicing, MEC for those large-sized tasks is done in the optimal manner by using the DASMO algorithm, which will search for the most optimal resource that satisfies the QoS requirements. Further, a joint network slicing method (Yuan, 2020) was developed for maintaining the QoE and QoS services with the more energy-efficient resources. This work utilised the sequential fixing process to obtain near-optimal solutions, and suboptimal solutions are solved by using the greedy approach. But still, energy-related issues are presented with respect to latency, network selection, capacity, service placement, and optimal allocation of resources. From the above survey, it is observed that the MEC performance is affected by the various threats as elaborated in Table 2.

4. OPEN ISSUES

In order to handle the expected exponential growth in demand for end-user resources from emerging applications, MECs must address the following main challenges addressed in the detailed literature:

Table 2. Classification of threats in MEC

Environment	Threats occurred
Network Infrastructure	rogue gateway, man-in-the-middle, and Denial of service
Edge data center	Rogue data center, service manipulation, privilege escalation, privacy leakage, and physical damage
Core infrastructures	Rogue infrastructure, service manipulation, and privacy leakage
Virtualization infrastructure	VM manipulation, privilege escalation, privacy leakage, misuse of resources, and denial of service
User devices	service manipulation and injection of information

Keeping computing and communication delays optimized, and (4) managing overall resource energy consumption. The following problems are studied in this paper:

4.1 How to Model the Resource Demands in Network Selection?

Understanding the resource demands and QoS requirements of applications in network selection is crucial when designing MEC algorithms capable of meeting the demands of emerging MEC network selection. The relevant parameters of an application need to be modeled to understand the underlying system dynamics, such as cost and performance dynamics. In (Manasrah 2018), a Hybrid GA-PSO Algorithm is used in Cloud Computing to give tasks to resources in the best way possible. The goal of the Hybrid GAPSO algorithm is to cut down on time and money as well as to spread the load of tasks that depend on each other across different resources in cloud computing environments. We have done something similar, but only used PSO in the edge layer. In (Omara, 2010), authors have used different kinds of genetic algorithms to solve various scheduling problems in the cloud. (Tien, 2019) also looks at modelling delay, energy loss, and cost. Our goal is to get the best response times by using an evolutionary algorithm for edge orchestration.

4.2 How to Plan Capacity and Service Placement for MECs?

The performance of MEC applications may have to satisfy key performance indicators specified by the end-users in service placement. For example, some applications may require that the average round-trip time be below a certain threshold, whereas others may impose requirements on tail latency according to the in-service placement. Tail latency is defined as the percentage of requests that can meet the desired latency requirement in a given time period to maintain QoS. Also, some applications can have a higher priority level than others, where the priority can be defined in terms of their QoS. The problem is how to allocate resources to such applications in order to minimize operational monetary costs for MECs while meeting application QoS. In the literature, this MEC problem is also called a service placement problem or an application provisioning problem. (Huang, 2018) suggested a parked vehicle edge computing (PVEC) architecture, where the resources of PVs that are not being used can be used to their full potential service placement. In the PVEC architecture, VEC servers look for resources from PVs that are available at the right time to divide up work, and they pay PVs for their help. If they have to, VEC servers can also take on the remaining work. So, VEC servers and PVs work together to process tasks in a way called "edge-edge collaboration". (Zhuojia Gu, 2021) came up with a different idea: a cloud-edge computing architecture that would allow horizontal and vertical collaborations and try to keep the total cost as low as possible. Horizontal collaboration means that nodes in the same tier can do offloading operations together, while vertical collaboration means that nodes in different tiers can do offloading operations together.

4.3 How to Optimally Allocate Resources to Applications in MECs with Reduced Latency?

The main factors that affect the MEC problem are the sensitivity of applications' requests with respect to resource types; space and time variation in their demand for resources; QoS requirements; and the capabilities of the available compute resources. For example, an application's requests (for resources) may be modeled by various parameters, such as arrival rate, service time, and data size and each of these parameters may be modeled using a random variable from an appropriate distribution. Similarly, the location of an end-user running an application may be modeled by a suitable latency distribution. The convex optimization method has been used a lot to solve the problem of reducing energy use (Wu, 2019), (Fan, 2021), (Xiang, 2021). Under the time constraint, the offloading policy has been optimized based on the channel gains and the amount of energy used by local computing. (Fan, 2020), focused on the number of users who could be offloaded, but they didn't optimize both radio and computing resources at the same time. The authors looked into a problem where radio and computing resources have to be shared in a non-orthogonal MEC in a heterogeneous network.

4.4 How to Handle the Overall Energy Consumption of Resource-Constrained IoT Devices in the MECs?

The MECs will host applications from a wide range of resource-constrained devices, making it complex to migrate or port applications from resource-constrained devices to the cloud and vice-versa. There is a need for a framework that can provide an abstraction of these heterogeneous, resource-constrained devices in order to allow for seamless development, deployment, and management of applications with reduced overall energy consumption. The framework should also make it easy to offload part of an application to the MECs to meet the application's QOS requirements. The framework should also support simple development, deployment, and management tools to facilitate the life-cycle management of applications. Most of the previously existing MEC-IoT based cloud scheduling approaches (Abbasi, 2021), (Bagher, 2021), (Bolettieri, 2021), (Cicconetti, 2020) never consider the interaction between different tasks and their influences on application security needs. To solve the problem of cloud load balancing issue, they have developed an advanced approach that includes both task security requirements and the interaction in between different tasks. In order to achieve better security and performance, they proposed an advanced heuristic scheme that depends upon the task's completion time and security needs. Furthermore, they have also introduced a new attack response technique to decrease several security threats in the cloud environment. Since years, there has been numerous research works carried out to develop an efficient and effective task scheduling algorithm. With the growing popularity of distributed systems, the number of challenges is also increasing. It is very difficult for the new researchers to understand the relationships in between various scheduling issues (Elazhary, 2019). These issues may influence the identification process of new research avenues. In this approach, they have presented an advanced classification scheme to solve these problems of scheduling in the distributed systems. They have performed survey on different scheduling schemes. It is very much essential to determine feasible solutions for satisfying the required service goals (Li, 2021). The above presented framework has the responsibility to permit various cloud users in order to improve the model of estimation before the actual execution process (Liyanage, 2021).

4.5 Traditional Issues of Meta-Heuristic Algorithms

Gravity search is a meta-heuristic optimization technique based on natural phenomena. This method is frequently employed to address problems with mobile edge load scheduling (Huang, 2019). It incorporates the fundamental ideas of Newton's gravitational law as well as others. This model is used to reduce transfer time and overall costs by planning cloud resources for virtual machines. Load balancing between the cloud resources and mobile edge devices allows optimum exploitation of the physical infrastructure while lowering run time (Mansouri, 2021). Quality of service (QoS) measures, such as response time, cost, performance, and resource utilization, can be improved by implementing load balancing in the cloud. A simplistic scheduling approach has a finite-sized multiple queue that takes on cloud services and allocates incoming tasks to adequate resources. The auto-scaling mechanism requires monitoring of task dynamics such as the time of waiting, time of waiting, number of queued tasks, and number of tasks awaiting. The quality of the self-scaling mechanism is reduced by frequent VM allocations and deallotment. The main goal of any cloud provider's resource provisioning is to get more resources used (Pasumpon Pandian, 2022).

4.6 Research Gaps

(Zou, 2021) proposed a hybrid service selection model in mobile edge computing. In this paper, a genetic algorithm-based service selection process is performed using the limited resource constraints. (Wang, 2021) proposed an edge-to-edge data storage service using an adaptive method. In this work, a hybrid data storage and service selection method is used to improve the task scheduling process of offloading data. Due to the lack of computing resources and data storage for largely offloaded tasks, a distributed resource constraint-based optimization model is required for the scheduling process. (Alkhalaileh, 2020) proposed a data-intensive offloading technique using mobile edge cloud computing. A non-linear integral optimization function is proposed for task offloading computation. This model is independent of resource constraints for the scheduling process. (Ali Shakarami, 2020) proposed survey on different offloading approaches in mobile edge computing. In this paper, machine learning-based offloading tasks are implemented and tested on different tasks. They classified supervised and unsupervised learning models for task offloading in mobile edge computing. The main issues with the supervised learning models include how to schedule a large number of tasks and being independent of resource constraints. The following table represents the summarization of tools and language used to implement the model for scheduling process.

5. PROPOSED OPTIMIZATION AND PERFORMANCE COMPARISON

In the section, an enhance version of PSO scheduling model (Chen 2020) is proposed using resource constraints. In this work, different resource constraints are used to optimize the offloading process. In the proposed approach, a new resource constraint based PSO model is proposed which is independent of the existing system. Also, this model has better computational runtime than the conventional approach. In Figure 5, mobile edge cloud systems are source devices to migrate or offload components during computation of the Amazon's EC2 cloud servers. Mobile devices need to select the wireless medium for component migration. Mobile devices, select available cloud wireless networks like 3G or Wi-Fi networks randomly. The data migration time between cloud upload, cloud download and energy statistics will be recorded.

In this model, we assume that the available cloud networks and the time of the application are not altered. When multiple networks are available on your local device, the face detection controller needs to choose the best network before deploying (e.g., Wi-Fi network or 3 G with the greatest frequency). A load-balancing algorithm that is efficient is needed to make the most of resources. An in-depth analysis of cloud computing most important aspects is carried out, as are comparisons of existing load balancing algorithms. As a method of partitioning the cloud, this technique focuses on the relationships between distinct cloud divisions. A person can be either overloaded, normal, or idle at any given time.

This system assumes each cloud server, can executes N virtual machines on the remote cloud environment. Let $< M_i, \mathrm{T}_i >$ represents the memory usage type of the ith representative virtual machine and $\mathrm{T}_i = \{\mathrm{t}_{ij}, t'_{ij}\}, \forall\, \mathrm{j}, \mathrm{t}_{ij} < t'_{ij}, t'_{ij} < t_{i(j+1)}$ represents server leasing periods and t_{ij}, t'_{ij} represents the start and ending runtime virtual machine instance. Let $\phi(M,T)$ represents cost computed by multiplying the total price with total runtime hours.

Key notations used in the Traditional PSO problem definition

VM_i : Represents with virtual machine instance.

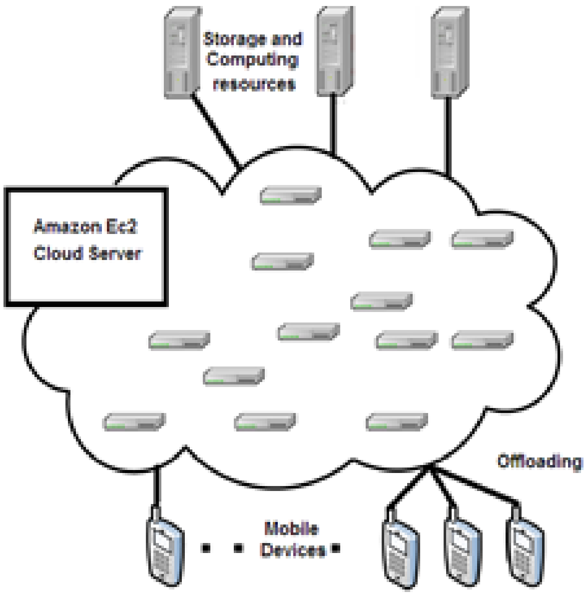
VM_{T_i} : Represents leasing periods of the i^{th} virtual machine.

$\phi(\mathrm{VM}_i, VM_{T_i})$: Represents a cloud price function for leasing a virtual machine instance.

Table 3. Represents the summarization of Tools and Platform

Reference	Tools	Platform
[84]	CLOUDSIM	Java
[85]	IFOGSIM	Java
[86]	IFOGSIM	Java
[87]	CLOUDSIM	Python

Figure 5. Mobile to cloud deploying model



T_k : kth computation task.

$I^r_{i,k}$ status function for offloading C_k task to i^{th} virtual machine

$I^l_{i,k}$ status function for offloading C_k task locally

$E^l_{i,k}$ status function for executing offloading T_k task locally

$E^r_{i,k}$ status function for executing offloading T_k task remotely

λ_k Maximum task execution time of the $k^{th}\, task$ and $\lambda_k \geq 1$

As per the PSO system, minimizing the task computation cost and resource allocation can be formulated as:

$$\min \sum_{i=1} \phi(\mathrm{VM}_i, VM_{T_i})$$

$$I^l_{i,k} + \sum_{i=1}^{N} I^r_{i,k} = 1 \text{ and}$$

$$E^l_{i,k} \,/\, \max\{E^l_{i,k} \,/\, I^l_{i,k}, \{E^r_{i,k} \,/\, I^r_{i,k}, \forall i\}\} \geq (1-\mathrm{r})\lambda$$

Where $r \in (0,1)$ arbitrary value selected by the server. (i.e.) if r=1, all tasks are executed locally otherwise executed remotely.

The local execution speed in the energy efficient task scheduling system can be determined using objective function

$$\max\{\sum_{k=1}^{N} E^{l}_{i,k} / \min\{E^{l}_{i,k} / I^{l}_{i,k}, \{\log(E^{r}_{i,k} / I^{r}_{i,k}), \forall i\}\}\} \tag{1}$$

$$I^{l}{}_{i,k} + \sum_{i=1}^{N} I^{r}{}_{i,k} = 1$$

Equation (1) gives the optimal response time of all tasks, for each value of k $\exists\ I^{r}{}_{i,k}, I^{l}{}_{i,k}$ equals to 1. If the number of tasks in the application increases, then the equation (1) fails to find the optimal results and worse leasing allocation.

Proposed resource constraint based PSO approach has following positive innovations than the conventional PSO method in mobile edge task scheduling process.

1. Proposed model has better multiple resource constraint scheduling process.
2. Current model has better computational runtime(ms) than the conventional approaches for multiple resource and task scheduling process.
3. Current model has better local power optimization than the conventional approaches.
4. Current model has better local optimization and global optimization for task to resource allocation process.

5.1 Experimental Results

Experimental results are simulated in CLOUDSIM and VanetMobiSim simulator for PSO and existing task and resourced scheduling process. Proposed and traditional methods are tested on VANET application-oriented data with different tasks and cloud instances. For simulation different parameters such as simulation area 2km, communication range 115m, uniform number of resources to each task, total number of service vehicles are 30. Proposed model is simulated with realistic united states census traffic map for service and task scheduling process. Here, Particle swarm optimization (PSO), Ant colony optimization(ACO) and Firefly optimization algorithms are implemented and tested on mobile cloud task scheduling process. In this simulation, small and medium virtual instances are used to test the performance of each approach for scheduling process. In this proposed simulation setup, a dynamic number of tasks, mobile nodes and edge servers are taken as input along with the resources. Here, resources are scheduled dynamically during the task scheduling process.

```
========== OUTPUT ==========
Task ID   STATUS   Data center ID   VM ID   Time     Start Time   Finish Time
    2       SUCCESS       4            4      168.02        0.5         168.52
    1       SUCCESS       3            3      386.49        0.5         386.99
    3       SUCCESS       5            5      389.45        0.5         389.95
    0       SUCCESS       2            2      504.38        0.5         504.88
    4       SUCCESS       6            6      515.51        0.5         516.01

The best fitness value: 671.3977218655303 Best makespan: 671.3977218655303
516.008
PSO finished!
shuting down server...i-058dca0ee6eda18f3
shuting down server...i-0de53a413cb39efd4
shuting down server...i-07c9a115ad2ae9f34
shuting down server...i-00232cfa2ba13b5d3
shuting down server...i-017bae323541079e3
Runtime(ms)   :244.61
```

```
========== OUTPUT ==========
Task ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0        SUCCESS        2            2      464.87       0.5         465.37
    4        SUCCESS        6            6      506.41       0.5         506.91
    1        SUCCESS        3            3      713.3       0.5         713.8
    3        SUCCESS        5            5      751.33       0.5         751.83
    2        SUCCESS        4            4      784.36       0.5         784.86

The best fitness value: 784.3622897466487 Best makespan: 784.3622897466487
784.862
QOS+IPSO finished!
shuting down server...i-058dca0ee6eda18f3
shuting down server...i-0de53a413cb39efd4
shuting down server...i-07c9a115ad2ae9f34
shuting down server...i-00232cfa2ba13b5d3
shuting down server...i-017bae323541079e3
Runtime(ms)  :339.19
```

In the above experimental result, different mobile tasks are scheduled using the optimized PSO model to the available resources and tasks. Each task and its computational time are computed for resource analysis.

5.2 Performance Metrics and Analysis

In this experimental result, different performance metrics such as energy consumption, runtime and task scheduling accuracy are used to perform the comparison of proposed model to the conventional models. Figure 2 represents the analysis of different resource constraint task scheduling approaches in mobile edge computing. In this fig 2, energy consumption of each mobile node for the task scheduling process is analyzed along with the conventional models. Figure 3, represent the performance analysis of different task scheduling models for runtime analysis. Also, figure 3 represents the comparative analysis of different task scheduling approaches for mobile task scheduling process in terms of efficiency. Figure 3 describes the task scheduling efficiency of proposed model to the conventional models on different mobile tasks.

6. CONCLUSION

In this paper, different meta-heuristic optimization models are used to test the scheduling process of different types of virtual machines and tasks. Since most of the conventional optimization models are

Figure 6. Comparative analysis of different resource constraint task scheduling approaches in mobile edge computing (energy consumption)

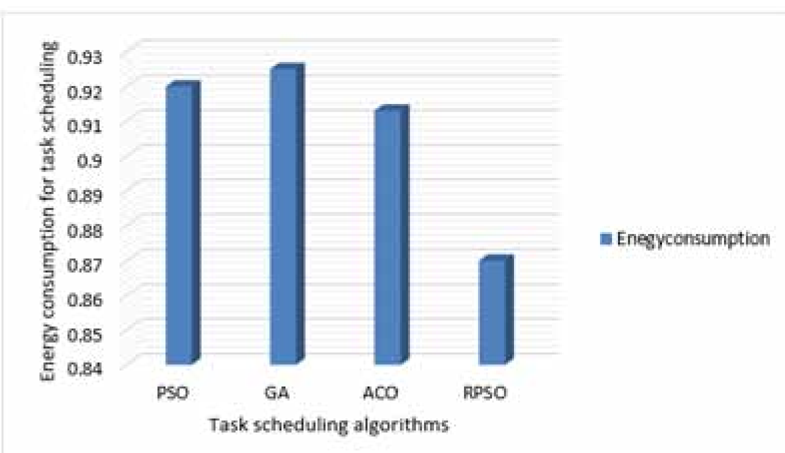


Figure 7. Comparative analysis of different resource constraint task scheduling approaches in mobile edge computing (runtime computation)

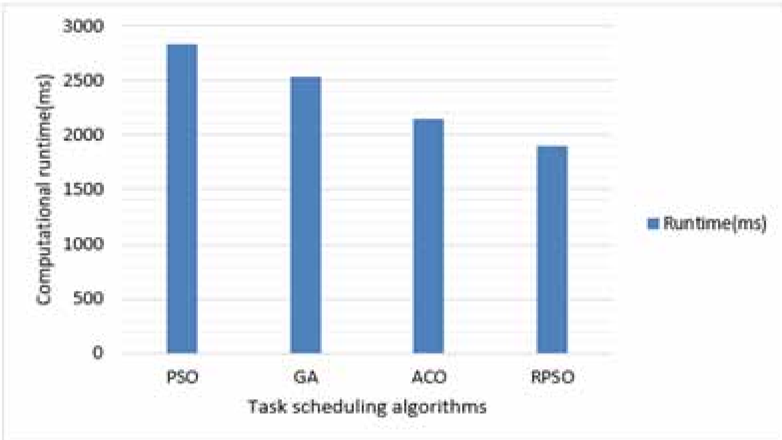
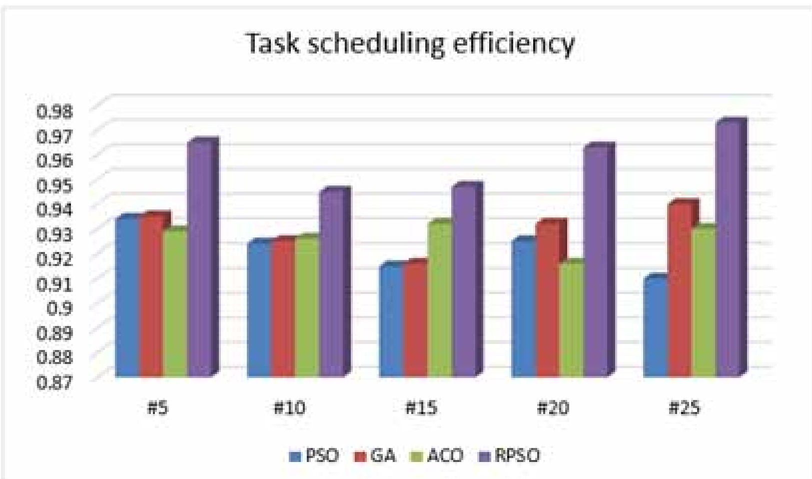


Figure 8. Performance analysis of proposed resource constraint PSO scheduling algorithm to the existing algorithms on different mobile tasks



difficult to schedule the tasks with different virtual machines. Also, as the size of the mobile edge offloading tasks increases, it is difficult to find the optimal load balancing meta-heuristic models due to the problem of local and global resource optimization process. In this work, different scheduling models are tested on different virtual machines for task scheduling process. In this work, a resource optimization based PSO approach is implemented to test the task scheduling process on mobile edge cloud computing environment using CLOUDSIM toolbox.

In this paper, a hybrid resource constraint-based particle swarm optimization is developed in order to improve the mobile edge computing task scheduling. In this paper, small and medium level cloud virtual machines are used to compute the local and global best PSO optimization parameters for scheduling process.

The following are the major enhancements that can be integrated to the traditional PSO model in order to achieve better QoS, scheduling time and resource scheduling process.

Table 4. Represents the challenges in MEC

Challenge	Description
Programmability	Network selection, session management and usability
Mobility	Service placement, seamless handoff, and connectivity
Distribution	Soft state, N-tier management, and cooperation
Resources & Tasks	Latency and overall energy consumption optimization, offloading, Task scheduling, and Resource allocation
Virtualization	context and container awareness, VM lifecycle
Infrastructure	Accountability, monitoring and interoperability

- Development of hybrid QoS measures that will be effectively utilized for the optimization of MEC's access network selection and service placement.
- Implementation of effective optimization approaches using meta-heuristic algorithms like particle swarm optimization, cuckoo search algorithm, and gray-wolf optimizer etc. to enhance the service of quality optimization through access network selection on network congestion.
- To develop a machine learning based mobile edge task classification for scheduling process in small to large scale virtual instances.

The above research gaps are used to improve the task and resource scheduling process in mobile edge computing for large scale applications. These research gaps significantly improve the latency, power and computational time for scheduling process.

REFERENCES

Abbasi, M., Mohammadi-Pasand, E., & Khosravi, M. R. (2021, March). Intelligent workload allocation in IoT–Fog–cloud architecture towards mobile edge computing. *Computer Communications*, *169*, 71–80. doi:10.1016/j.comcom.2021.01.022

Ahn, J., Lee, J., Park, S., & Park, H.-S. (2019, April). Power efficient clustering scheme for 5G mobile edge computing environment. *Mobile Networks and Applications*, *24*(2), 643–652. doi:10.1007/s11036-018-1164-2

Al-Badarneh, J., Jararweh, Y., Al-Ayyoub, M., Fontes, R., Al-Smadi, M., & Rothenberg, C. (2018, October). Cooperative mobile edge computing system for VANET based software-defined content delivery. *Computers & Electrical Engineering*, *71*, 388–397. doi:10.1016/j.compeleceng.2018.07.021

Alameddine, H. A., Sharafeddine, S., Sebbah, S., Ayoubi, S., & Assi, C. (2019, March). Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. *IEEE Journal on Selected Areas in Communications*, *37*(3), 668–682. doi:10.1109/JSAC.2019.2894306

Alfakih, T., Hassan, M. M., & Al-Razgan, M. (2021). Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing. *IEEE Access: Practical Innovations, Open Solutions*, *9*, 167503–167520. doi:10.1109/ACCESS.2021.3134941

Alkhalaileh, M., Calheiros, R. N., Nguyen, Q. V., & Javadi, B. (2020, October). Data-intensive application scheduling on Mobile Edge Cloud Computing. *Journal of Network and Computer Applications*, *167*, 102735. doi:10.1016/j.jnca.2020.102735

Babou, C. S. M., Fall, D., Kashihara, S., Taenaka, Y., Bhuyan, M. H., Niang, I., & Kadobayashi, Y. (2020). Hierarchical load balancing and clustering technique for home edge computing. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 127593–127607. doi:10.1109/ACCESS.2020.3007944

Bagher, K., Khalil, I., Alabdulatif, A., & Atiquzzaman, M. (2021, April). EdgeSOM: Distributed hierarchical edge-driven IoT data analytics framework. *Computer Communications*, *172*, 64–74. doi:10.1016/j.comcom.2021.02.021

Best-Rowden, L., & Jain, A. K. (2018, January). Longitudinal study of automatic face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(1), 148–162. doi:10.1109/TPAMI.2017.2652466 PMID:28092523

Bi, J., Yuan, H., Duanmu, S., Zhou, M., & Abusorrah, A. (2020). Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization. *IEEE Internet of Things Journal*, *8*(5), 3774–3785. doi:10.1109/JIOT.2020.3024223

Bolettieri, S., Bruno, R., & Mingozzi, E. (2021, May). Application-aware resource allocation and data management for MEC-assisted IoT service providers. *Journal of Network and Computer Applications*, *181*, 103020. doi:10.1016/j.jnca.2021.103020

Chen, C., Chen, L., Liu, L., He, S., Yuan, X., Lan, D., & Chen, Z. (2020, January). Delay-Optimized V2V-Based Computation Offloading in Urban Vehicular Edge Computing and Networks. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 18863–18873. doi:10.1109/ACCESS.2020.2968465

Chen, X. (2014). 2316834. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 26*(4), 974–983. <ALIGNMENT.qj></ALIGNMENT>10.1109/TPDS

Chen, X., Jiao, L., Li, W., & Fu, X. (October 2016). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking, 24*(5), 2795–2808. . IEEE, ACM.10.1109/TNET.2015.2487344

Chen, Y.-T., & Liao, W. (May 2019). Mobility-aware service function chaining in 5G wireless networks with mobile edge computing. In *Proceedings of the IEEE International Conference Commun* (pp. 1–6). ICC. doi:10.1109/ICC.2019.8761306

Chou, L.-D., Tseng, C.-W., Huang, Y.-K., Chen, K.-C., Ou, T.-F., & Yen, C.-K. (October 2016). A security service on-demand architecture in SDN. In *Proceedings of the International Conference Inf. Commun. Tech. Converg.* (ICTC) (pp. 287–291). IEEE. doi:10.1109/ICTC.2016.7763487

Cicconetti, C., Conti, M., & Passarella, A. (2020, May). Uncoordinated access to serverless computing in MEC systems for IoT. *Computer Networks*, *172*, 107184. doi:10.1016/j.comnet.2020.107184

Cominardi, L., Deiss, T., Filippou, M., Sciancalepore, V., Giust, F., & Sabella, D. (2020, June). MEC support for network slicing: Status and limitations from a standardization viewpoint. *IEEE Communications Standards Magazine*, *4*(2), 22–30. doi:10.1109/MCOMSTD.001.1900046

Dai, Y., Xu, D., Maharjan, S., & Zhang, Y. (2019, June). Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet of Things Journal*, *6*(3), 4377–4387. doi:10.1109/JIOT.2018.2876298

Deng, M., Tian, H., & Lyu, X. (May 2016). Adaptive sequential offloading game for multi-cell mobile edge computing. In *Proceedings of the 23rd International Conference Telecommunication (ICT)* (pp. 1–5). IEEE. . doi:10.1109/ICT.2016.7500395

Do, T. V., Do, N. H., Nguyen, H. T., Rotter, C., Hegyi, A., & Hegyi, P. (2019, May). Comparison of scheduling algorithms for multiple mobile computing edge clouds. *Simulation Modelling Practice and Theory*, *93*, 104–118. doi:10.1016/j.simpat.2018.10.005

Elazhary, H. (2019, February). Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. *Journal of Network and Computer Applications*, *128*, 105–140. doi:10.1016/j.jnca.2018.10.021

Fan, W., Han, J., Yao, L., Wu, F., & Liu, Y. (2020, November). Latency-energy optimization for joint WiFi and cellular offloading in mobile edge computing networks. *Computer Networks*, *181*, 107570. doi:10.1016/j.comnet.2020.107570

Feng, J., Pei, Q., Yu, F. R., Chu, X., Du, J., & Zhu, L. (2020, July). Dynamic network slicing and resource allocation in mobile edge computing systems. *IEEE Transactions on Vehicular Technology*, *69*(7), 7863–7878. doi:10.1109/TVT.2020.2992607

Forti, S., Paganelli, F., & Brogi, A. (2022). Probabilistic QoS-aware placement of VNF chains at the edge. *Theory and Practice of Logic Programming*, *22*(1), 1–36. doi:10.1017/S1471068421000016

Gu, Z., Lu, H., Zou, C. (2021). Horizontal and Vertical Collaboration for VR Delivery. In *MEC-enabled small-cell networks*.

Han, Y., Guo, D., Cai, W., Wang, X., & Leung, V. (2020). Virtual machine placement optimization in mobile cloud gaming through QoE-oriented resource competition. *IEEE Transactions on Cloud Computing*, 1–1. IEEE. 10.1109/TCC.2020.3002023

Huang, B., Li, Z., Tang, P., Wang, S., Zhao, J., Hu, H., Li, W., & Chang, V. (2019, August). Security modeling and efficient computation offloading for service workflow in mobile edge computing. *Future Generation Computer Systems*, *97*, 755–774. doi:10.1016/j.future.2019.03.011

Huang, C.-M., Chiang, M.-S., Dao, D.-T., Su, W.-L., Xu, S., & Zhou, H. (2018). V2 V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 17741–17755. doi:10.1109/ACCESS.2018.2820679

Huang, X., Yu, R., Liu, J., & Shu, L. (2018). Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 66649–66663. doi:10.1109/ACCESS.2018.2879578

Kaur, K., Garg, S., Aujla, G. S., Kumar, N., Rodrigues, J. J. P. C., & Guizani, M. (2018, February). Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay. *IEEE Communications Magazine*, *56*(2), 44–51. doi:10.1109/MCOM.2018.1700622

Kourtis, M.-A., Blanco, B., Perez-Romero, J., Makris, D., Mcgrath, M. J., Xilouris, G., Munaretto, D., Solozabal, R., Sanchoyerto, A., Giannoulakis, I., Kafetzakis, E., Riccobene, V., Jimeno, E., Kourtis, A., Ferrus, R., Liberal, F., Koumaras, H., Kostopoulos, A., & Chochliouros, I. (2019). A cloud-enabled small cell architecture in 5G networks for broadcast/multicast services. *IEEE Transactions on Broadcasting*, *65*(2), 414–424. doi:10.1109/TBC.2019.2901394

Ksentini, A., & Frangoudis, P. A. (2020, March). Toward slicing-enabled multiaccess edge computing in 5G. *IEEE Network*, *34*(2), 99–105. doi:10.1109/MNET.001.1900261

Ksentini, A., Taleb, T., & Chen, M. (2014, June). A Markov decision process-based service migration procedure for follow me cloud. *Proceedings of the IEEE*, *ICC*, 1350–1354.

Kumar, K., & Lu, Y.-H. (2010, April). Cloud computing for mobile users: Can offloading computation save energy? *Computer*, *43*(4), 51–56. doi:10.1109/MC.2010.98

Lakhan, A., Abed Mohammed, M., Rashid, A. N., Kadry, S., Hameed Abdullkareem, K., Jan, N., Martinek, R., & Imran, R. (2022, May). Restricted Boltzmann machine Assisted Secure Serverless Edge System for Internet of Medical Things. *IEEE Journal of Biomedical and Health Informatics*, 1. doi:10.1109/JBHI.2022.3178660 PMID:35635827

Lakhan, A., Elhoseny, M., Abed Mohammed, M., & Mustafa, M. J. (2022). SFDWA: Secure and Fault-Tolerant Aware Delay Optimal Workload Assignment Schemes in Edge Computing for Internet of Drone Things Applications. *Wireless Communications and Mobile Computing*. Hindawi. 10.1155/2022/5667012

Lakhan, A., Mohammed, M. A., Rashid, A. N., Kadry, S., & Abdulkareem, K. H. (2022). Deadline aware and energy-efficient scheduling algorithm for fine-grained tasks in mobile edge computing. *IJWGS*, *18*(2), 168–193. doi:10.1504/IJWGS.2022.121935

Leivadeas, A., Kesidis, G., Ibnkahla, M., & Lambadaris, I. (2019, March). VNF placement optimization at the edge and cloud. *Future Internet*, *11*(3), 69. doi:10.3390/fi11030069

Li, D., Hong, P., Xue, K., & Pei, J. (2019, April). Virtual network function placement and resource optimization in NFV and edge computing enabled networks. *Computer Networks*, *152*, 12–24. doi:10.1016/j.comnet.2019.01.036

Li, G., Zhou, H., Feng, B., Li, G., Li, T., Xu, Q., & Quan, W. (2017). Fuzzy theory based security service chaining for sustainable mobile-edge computing. *Mobile Information Systems*, *2017*, 1–13. doi:10.1155/2017/8098394

Li, H., Shou, G., Hu, Y., & Guo, Z. (March 2016). Mobile edge computing: Progress and challenges. In *Proceedings of the 4th IEEE International Conference Mobile Cloud Computability, Services, Eng. (MobileCloud)* (pp. 83–84). IEEE.

Li, J. Q., Song, M. X., Wang, L., Duan, P. Y., Han, Y. Y., Sang, H. Y., & Pan, Q. K. (2020, June). Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs. *IEEE Transactions on Cybernetics*, *50*(6), 2425–2439. doi:10.1109/TCYB.2019.2943606 PMID:31603832

Li, X., Zhao, L., Yu, K., Aloqaily, M., & Jararweh, Y. (2021, May). A cooperative resource allocation model for IoT applications in mobile edge computing. *Computer Communications*, *173*, 183–191. doi:10.1016/j.comcom.2021.04.005

Liu, F., Shu, P., Jin, H., Ding, L., & Yu, J. (2013, June). Gearing resource poor mobile devices with powerful clouds: Architectures, challenges, and applications. *IEEE Wireless Communications*, *20*(3), 14–22. doi:10.1109/MWC.2013.6549279

Liu, F., Shu, P., & Lui, J. C. S. (2015, November). AppATP: An energy conserving adaptive mobile-cloud transmission protocol. *IEEE Transactions on Computers*, *64*(11), 3051–3063. doi:10.1109/TC.2015.2401032

Liyanage, M., Porambage, P., Ding, A. Y., & Kalla, A. (2021, June). Driving forces for multi-access edge computing (MEC) IoT integration in 5G. *ICT Express*, *7*(2), 127–137. doi:10.1016/j.icte.2021.05.007

Ma, S., Song, S., Zhao, J., Zhai, L., & Yang, F. (2020). Joint network selection and service placement based on particle swarm optimization for multi-access edge computing. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 160871–160881. doi:10.1109/ACCESS.2020.3020935

Ma, S., Song, S., Zhao, J., Zhai, L., & Yang, F. (2020). Joint network selection and service placement based on particle swarm optimization for multi-access edge computing. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 160871–160881. doi:10.1109/ACCESS.2020.3020935

Manasrah, A. M., & Ba Ali, H. B. (2018, January). Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. *Wireless Communications and Mobile Computing*, 1–16. doi:10.1155/2018/1934784

Mansouri, Y., & Babar, M. A. (2021, April). A review of edge computing: Features and resource virtualization. *Journal of Parallel and Distributed Computing*, *150*, 155–183. doi:10.1016/j.jpdc.2020.12.015

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys and Tutorials, 19*(4), 2322–2358. 10.1109/COMST.2017.2745201

Mitsis, G., Apostolopoulos, P. A., Tsiropoulou, E. E., & Papavassiliou, S. (2019, May). Intelligent dynamic data offloading in a competitive mobile edge computing market. *Future Internet*, *11*(5), 118. doi:10.3390/fi11050118

Nicholson, A. J., Chawathe, Y., Chen, M. Y., Noble, B. D., & Wetherall, D. (2006). Improved access point selection. In *Proceedings of the 4th International Conference Mobile System.* Appliance Services (pp. 233–245).

Omara, F. A., & Arafa, M. M. (2010, January). Genetic algorithms for task scheduling problem. *Journal of Parallel and Distributed Computing*, *70*(1), 13–22. doi:10.1016/j.jpdc.2009.09.009

Pasumpon Pandian, A., Ntalianis, K., & Palanisamy, R. (2022, May). Introduction to the special section on Artificial Intelligence-driven Mobile Edge Computing (VSI-aime). *Computers & Electrical Engineering*, *100*, 108070. doi:10.1016/j.compeleceng.2022.108070

Pavlos, A., Tsiropoulou, E. E., & Papavassiliou, S. (2018). Game-theoretic learning-based QoS satisfaction in autonomous mobile edge computing. In *Proceedings of the Global Inf. Infrastruct. Netw. Symposium (GIIS)* (pp. 1–5). IEEE. doi:10.1109/GIIS.2018.8635770

Peng, H., Ye, Q., & Shen, X. S. (2019, August). SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach. *IEEE Wireless Communications*, *26*(4), 156–162. doi:10.1109/MWC.2019.1800371

Piao, J. T., & Yan, J. (November 2010). A network-aware virtual machine placement and migration approach in cloud computing. In *Proceedings of the 9th International Conference Grid Cloud Computability* (pp. 87–92). IEEE. doi:10.1109/GCC.2010.29

Ruiz, L., Durán, R., de Miguel, I., Khodashenas, P., Pedreño-Manresa, J.-J., Merayo, N., Aguado, J., Pavón-Marino, P., Siddiqui, S., Mata, J., Fernández, P., Lorenzo, R., & Abril, E. (2018, December). A genetic algorithm for VNF provisioning in NFV-enabled Cloud/MEC RAN architectures. *Applied Sciences (Basel, Switzerland)*, *8*(12), 2614. doi:10.3390/app8122614

Sanguanpuak, T. (2018). Network slicing with mobile edge computing for micro-operator networks. in *beyond 5G 21st International Symposium on Wireless Personal Multimedia Communications (WPMC).* IEEE Publications.

Sarrigiannis, I., Kartsakli, E., Ramantas, K., Antonopoulos, A., & Verikoukis, C. (2018). Application and network VNF migration in a MEC enabled 5G architecture. In *Proceedings of the IEEE 23rd Interface Workshop Computability Aided Modeling Design Commun. Links Netw. (CAMAD)* (pp. 1–6). IEEE.

Sechkova, T., Paolino, M., & Raho, D. (2018). Virtualized infrastructure managers for edge computing: Openvim and openstack comparison *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB).* IEEE Publications.

Shah, S. D. A., Gregory, M. A., Li, S., & Fontes, R. D. R. (2020). SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 77459–77469. doi:10.1109/ACCESS.2020.2990292

Shakarami, A., Ghobaei-Arani, M., & Shahidinejad, A. (2020, December). A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. *Computer Networks*, *182*, 107496. doi:10.1016/j.comnet.2020.107496

Solozabal, R., Blanco, B., Fajardo, J. O., Taboada, I., Liberal, F., Jimeno, E., & Lloreda, J. G. (2017). Design of virtual infrastructure manager with Novel VNF placement features for edge clouds in 5G. In G. Boracchi, L. Iliadis, C. Jayne, & A. Likas (Eds.), *Engineering applications of neural networks(communications in computer and information science)* (pp. 669–679). Springer. doi:10.1007/978-3-319-65172-9_56

Son, J., & Buyya, R. (2019, June). Latency-aware virtualized network function provisioning for distributed edge clouds. *Journal of Systems and Software*, *152*, 24–31. doi:10.1016/j.jss.2019.02.030

Srivatsa, M., Ganti, R., Wang, J., & Kolar, V. (2013). 'Map matching: Facts and myths,' in. Proceedings of the 21st ACM SIGSPATIAL International Conference Advance Geographic Inf. Syst. (pp. 484–487). doi:10.1145/2525314.2525466

Sun, G., Li, Y., Li, Y., Liao, D., & Chang, V. (2018, August). Low-latency orchestration for workflow-oriented service function chain in edge computing. *Future Generation Computer Systems*, *85*, 116–128. doi:10.1016/j.future.2018.03.018

Sun, S., Luo, C., & Chen, J. (2017, July). A review of natural language processing techniques for opinion mining systems. *Information Fusion*, *36*, 10–25. doi:10.1016/j.inffus.2016.10.004

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys and Tutorials, 19*(3), 1657–1681. 10.1109/COMST.2017.2705720

Tomaszewski, L., Kukliński, S., & Kołakowski, R. (2020). A new approach to 5G and MEC integration. In I. Maglogiannis, L. Iliadis & E. Pimenidis (Eds.). *Proceedings of the Artif. Intell. Appl. Innov. AIAI IFIP WG Int. Workshops in Advances in Information and Communication Technology* (pp. 15–24). Springer.

Tran, T. X., Hajisami, A., Pandey, P., & Pompili, D. (2017, April). Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine*, *55*(4), 54–61. doi:10.1109/MCOM.2017.1600863

Tun, Y. K., Alsenwi, M., Pandey, S. R., Zaw, C. W., & Hong, C. S. (2019, September). Energy efficient multi-tenant resource slicing in virtualized multi-access edge computing. In *20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, (pp. 1–4). IEEE Publications. doi:10.23919/APNOMS.2019.8892978

Vladyko, A., Khakimov, A., Muthanna, A., Ateya, A. A., & Koucheryavy, A. (2019, June). Distributed edge computing to assist ultra-lowlatency VANET applications. *Future Internet*, *11*(6), 128. doi:10.3390/fi11060128

Wang, J., Chen, H., Zhou, F., Sun, M., Huang, Z., & Zhang, Z. (2021, July). A-DECS: Enhanced collaborative edge–edge data storage service for edge computing with adaptive prediction. *Computer Networks*, *193*, 108087. doi:10.1016/j.comnet.2021.108087

Wang, L., Jiao, L., He, T., Li, J., & Muhlhauser, M. (April 2018). Service entity placement for social virtual reality applications in edge computing. In *Proceedings of the IEEE Infocom* (pp. 468–476). IEEE. doi:10.1109/INFOCOM.2018.8486411

Wang, Q., Alcaraz-Calero, J., Ricart-Sanchez, R., Weiss, M. B., Gavras, A., Nikaein, N., Vasilakos, X., Giacomo, B., Pietro, G., Roddy, M., Healy, M., Walsh, P., Truong, T., Bozakov, Z., Koutsopoulos, K., Neves, P., Patachia-Sultanoiu, C., Iordache, M., Oproiu, E., & Lomba, C. (2019, June). Enable advanced QoS-aware network slicing in 5G networks for slice-based media use cases. *IEEE Transactions on Broadcasting*, *65*(2), 444–453. doi:10.1109/TBC.2019.2901402

Wang, S., Urgaonkar, R., Zafer, M., He, T., Chan, K., & Leung, K. K. (May 2015). Dynamic service migration in mobile edge-clouds. In *Proceedings of the IFIP/IEEE Network Conference* (pp. 1–9). IEEE.

Wang, C., He, Y., Yu, F. R., Chen, Q., & Tang, L. (2018). Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges. *IEEE Communications Surveys and Tutorials, 20*(1), 7–38. 10.1109/COMST.2017.2758763

Wu, H., Lyu, X., & Tian, H. (2019, December). Online optimization of wireless powered mobile-edge computing for heterogeneous industrial Internet of things. *IEEE Internet of Things Journal*, *6*(6), 9880–9892. doi:10.1109/JIOT.2019.2932995

Xiang, B., Elias, J., Martignon, F., & Di Nitto, E. (2021, November). Resource calendaring for Mobile Edge Computing: Centralized and decentralized optimization approaches. *Computer Networks*, *199*, 108426. doi:10.1016/j.comnet.2021.108426

Yala, L., Frangoudis, P. A., & Ksentini, A. (December 2018). Latency and availability driven VNF placement in a MEC-NFV environment. In *Proceedings of the IEEE Global Commun. Conference (GLOBECOM)* (pp. 1–7). IEEE. doi:10.1109/GLOCOM.2018.8647858

Yang, B., Chai, W. K., Xu, Z., Katsaros, K. V., & Pavlou, G. (2018, March). Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications. *IEEE eTransactions on Network and Service Management*, *15*(1), 475–488. doi:10.1109/TNSM.2018.2790081

Yang, L., Cao, J., Tang, S., Li, T., & Chan, A. T. S. Jun. 2012. A framework for partitioning and execution of data stream applications in mobile cloud computing. In *Proceedings of the IEEE 5th International Conference Cloud Computability*, (pp. 794–802). IEEE. doi:10.1109/CLOUD.2012.97 doi:10.1109/CLOUD.2012.97

Yang, Q., Chen, W.-N., Yu, Z., Gu, T., Li, Y., Zhang, H., & Zhang, J. (2017, April). Adaptive multimodal continuous ant colony optimization. *IEEE Transactions on Evolutionary Computation*, *21*(2), 191–205. doi:10.1109/TEVC.2016.2591064

You, Q., & Tang, B. (2021). Efficient task offloading using particle swarm optimization algorithm in edge computing for industrial internet of things. *Journal of Cloud Computing*, *10*(1), 1–11.

Yuan, H., & Zhou, M. (2020). Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems. *IEEE Transactions on Automation Science and Engineering*, *18*(3), 1277–1287. doi:10.1109/TASE.2020.3000946

Zhang, Y., Liu, Y., Zhou, J., Sun, J., & Li, K. (2020). Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing. *Future Generation Computer Systems*, *112*, 148–161. doi:10.1016/j.future.2020.05.025

Zhu, H., & Huang, C. (2018). EdgePlace: Availability-aware placement for chained mobile edge applications. *Transactions on Emerging Telecommunications Technologies*, *29*(11), e3504. doi:10.1002/ett.3504

Zou, G., Qin, Z., Deng, S., Li, K.-C., Gan, Y., & Zhang, B. (2021, April). Towards the optimality of service instance selection in mobile edge computing. *Knowledge-Based Systems*, *217*, 106831. doi:10.1016/j.knosys.2021.106831

I B. Teja Sree is currently pursuing a Ph.D. degree in computer science and engineering from KL University, Vaddeswaram. Sree has received a B. Tech and M. Tech degrees from the Sagi Rama Krishnam Raju Engineering college, affiliated to Andhra University, in 2014 and 2017. Sree is currently working as an Assistant professor in Sagi Rama Krishnam Raju Engineering College, affiliated to JNTU Kakinada. Sree's current research interests include Mobile Edge Computing.