JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Review on Task Scheduling in Edge Cloud Computing using Optimization Algorithms

[1]K.Vinothkumar, [2]Dr. D. Maruthanayagam
[1]Research Scholar, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India
[2] Head/Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India.

Abstract: Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of **IoT** services. Cloud computing to improve network efficiency and satisfy the internet requirement with connected devices increasing by moving computation, control and data storage into the cloud. However, cloud computing faces some challenges to some more stringent performance, such as latency and bandwidth, which are required by many application services. ***By combining the benefits of cloud computing and edge computing, edge-cloud computing is one of the most promising ways to address all of above problems*** for improving the battery lifetime and application performance for user devices. Edge-cloud computing performs each task on a user device, an edge or a cloud, which can provide better computing performance and transmission performance compared with edge computing or cloud computing in overall. Improved privacy of user data in the Edge-Cloud collaboration Task scheduling in cloud computing works based on the current information of tasks and resources in accordance with a certain strategy in order to establish an appropriate mapping relationship of tasks to appropriate resources. ***In this paper we discussed various research related with Edge-Cloud Collaborative Scheduling of Computing Tasks using Optimization algorithms.***

Keywords: Task Scheduling, Edge Cloud Computing, Optimization Algorithms, Cloud Data Centers, Genetic Algorithm, Cat Swarm Optimization, Multi-Objective Ant Colony and Multi Objective Particle Swarm Optimization

I. INTRODUCTION

In a cloud computing paradigm, users can rely on extremely rich storage and computing resources of a cloud computing center to expand the computing and storage power of devices, and achieve the rapid processing of computing intensive tasks. Yet there are some disadvantages in the device cloud collaboration mode, such as incurring high transmission delay and pushing network bandwidth requirement to the limit. In order to solve the problems of cloud computing for data processing, edge computing is put forward to provide desired computing services for users by using computing, network, storage and other resources on edge, that is near a physical entity or data source [1]. Edge computing moves the services and functions originally located in the cloud to the proximity of users, which integrates the cloud computing platform and the network to provide powerful computing, storage, networking, and communication capacity at the edge of the network. Compared with cloud computing, some applications of users in edge computing can be processed on an edge server near intelligent devices, thus significantly reducing data transmission delay and network bandwidth load required in edge-cloud collaboration [2]. Edge-cloud collaborative services are split into several heterogeneous computing tasks, where the latency-sensitive computing tasks are offloaded to edge servers, while the computing tasks that are latency-insensitive and computational intensive are served by cloud Data Centers (DC) [3].

The traditional scheduling strategies of edge computing tasks are to offload all computing-intensive tasks of edge devices to an edge server for processing. However, it may result in the waste of computing and storage sources in edge devices and cloud computing centers. In addition, many devices may access an edge server at the same time period. As a result, the server may face too many computing tasks, thus resulting in a long queue of tasks. This increases the completion time of all queued tasks, even causing the processing delay of tasks in the edge server to exceed that at the edge devices. On the other hand, many edge devices may be idle, resulting in a waste of their computing resources; and resource-rich cloud centers may be underutilized. **To solve the above problems, we can combine a cloud center, edge servers and edge devices** together to efficiently handle the computing tasks of edge devices via task offloading. According to the computing tasks' characteristics, optimization objectives and system status, we should utilize the computing and storage resources of a cloud center, edge servers and edge devices, and schedule computing tasks to them for processing on demand. It can effectively reduce the load of edge servers and improve the utilization of resources, and reduce the average completion time of computing tasks in a system [4][5].

Cloud computing can deliver the required computational and storage resources to the IoT, but this incurs high traffic and long delays. Therefore, there is a need for more computational and storage resources at the IoT networks' edge to offload. Computation offloading was first introduced as a promising paradigm for augmenting the computing capabilities of devices by allowing them to use remote cloud servers for performing their computational tasks [6]. Computation offloading to remote cloud servers may indeed (really) accelerate the execution of computational tasks, but due to high communication delays it may not be able to meet the requirements of latency sensitive applications. In order to meet the extremely low latency requirements of emerging delay sensitive applications, novel paradigms of computation offloading propose bringing computing and storage resources closer to the end users, that is, to the network edge [7]. Generally, there are three kinds of offloading policies aiming at minimal delay, minimal energy and balance between delay and energy. In order to simultaneously consider the three optimization goals of task completion time, energy consumption, and load balancing during task scheduling in an edge computing environment with limited node computing power, a multi-node optimization-based as well as multi-node task scheduling technology based on multi-objective optimization. In this next section we discussed **various research related with Edge Cloud based multi-objective optimization algorithms.**

II. LITERATURE REVIEW

XueLi Yao *et al.*,[8] established a cloud workflow scheduling model based on completion time and execution cost, then proposes a MOEA/D algorithm based on local search and weight vector adjustment, whose results show that in solving the cloud workflow scheduling model based on completion time and execution cost. Firstly, by deeply analyzing the cloud workflow scheduling process and its characteristics, the cloud workflow scheduling model based on completion time and execution cost is established. And followed by this, MOEA/D algorithm based on local search and weight vector adjustment is proposed and applied to cloud workflow scheduling problem. The experimental results show that the proposed algorithm has better effect than MOEA/D algorithm and NSGA-II algorithm for most actual workflow scheduling schemes. At the same time, a group of uniformly distributed Pareto dominant solutions are obtained, which can effectively provide decision support for cloud workflow scheduling problem. In the future research of workflow scheduling problem, research can add the scheduling scheme of traditional heuristic method to the initialization population on the basis of this research. Meanwhile, considering that this research adopts the random crossover and mutation evolutionary operator, researcher can use the heuristic crossover and mutation operator to accelerate the convergence of the algorithm in the future.

Asad Mahmood *et al.*,[9] researcher turned in optimization problem. The objective is to minimize the task duration by optimal allocation of the resources like local and edge computational capabilities, transmission power, and optimal task segmentation. For optimal allocation of resources, an algorithm name Estimation of Optimal Resource Allocator (EORA) is designed to optimize the function by keeping track of statistics of each candidate of the population. Using EORA, a comparative analysis of the hybrid approach (partial offloading) and edge computation only is performed. Results reveal the fundamental trade-off between both of these models. Simultaneously, the impact of devices' computational capability, data volume, and computational cycles requirement on task segmentation is analyzed. In this research presented a convex optimization problem is formulated to minimize the task's computational time by optimally allocating resources among local devices and mobile edge cloud. **Two models name edge computation and hybrid approach: partial offloading scheme is considered.** Besides this, it also reveals the fundamental trade-off between edge computation and a hybrid approach. Simulation results reveal that researcher proposed model is more efficient as compared to edge computation. Furthermore, research analyze the impact of task segmentation on systems' performance and find that optimal segmentation can effectively reduce the tasks' computational time compared to fixed or random segmentation. Simulation results demonstrate that the hybrid approach: partial offloading scheme reduces the task's computation time and outperforms edge computing only.

Mahbuba Afrin *et al.*,[10] researcher addressed simultaneous optimization of makespan, energy consumption and cost while allocating resources for the tasks of a robotic workflow. As a use case, this research considers resource allocation for the robotic workflow of emergency management service in smart factory. This work design an Edge Cloud based multi-robot system to overcome the limitations of remote Cloud based system in exchanging delay sensitive data. The resource allocation for robotic workflow is modeled as a constrained multi-objective optimization problem and **it is solved through a multi-objective evolutionary approach, namely, NSGA-II algorithm.** This research have redesigned the NSGA-II algorithm by defining a new chromosome structure, pre-sorted initial population and mutation operator. It is further augmented by selecting the minimum distant solution from the non-dominated front to the origin while crossing over the chromosomes. The experimental results based on synthetic workload demonstrate that our augmented NSGA-II algorithm outperforms the state-of-the-art works by at least 18% in optimizing makespan, energy and cost attributes on various scenarios. Research proposed an Edge Cloud robotic framework that is used for allocating computing resources for multi-robotic workflow of smart factory applications. To overcome the limitations of remote Cloud, a middle tier Edge Cloud is incorporated between local robots and remote Cloud that satisfies inter-task dependency within a workflow. The resource allocation problem is formulated as a constrained multi-objective optimization problem that optimizes the makespan for completing the tasks, the energy consumption of resources and the total cost for executing the tasks simultaneously. To solve this problem, the classical multi-objective evolutionary approach, namely NSGA-II is redesigned in his system with pre-sorted population and the fittest chromosome selection during crossover. The researcher proposed approach is compared with existing optimization approaches (benchmark NSGA-II, MOPSO, SPEA2, PAES) and it gives better performance in different simulation scenarios. Research is confident that our solution can be applied in other real-world scenarios in future. This work can also be extended to deal with the dynamic environment (mobility of robots, link failures, and limited bandwidth) of smart factory. Utilization of Edge resources for real-time applications will be a good extension of our work as well.

Zheng Shi *et al.*,[11] proposed multi-node task scheduling technology, a multi-objective optimization model is established, while considering the impact of completion time, energy consumption and load balancing on task scheduling. The task scheduling problem is transformed into a bidding model, and the offloading location of subtasks is determined in real time to meet the requirements of delay-sensitive tasks. Finally, simulation experiments are used to obtain the availability of the technology for multi-node task scheduling, which provides new ideas for task scheduling in edge computing. A multi-objective optimization unified model is used to optimize the bidding model, and an adaptive balanced scheduling strategy is used to optimize the bidding (request) strategy in the bidding process so that the tasks are more evenly scheduled to each node. Experiments were conducted in the CloudSim simulation environment. The experiments proved the effectiveness of the algorithm, and the complexity satisfies the constraints of running on edge devices. **In actual situations, task scheduling is more complicated**. In the future, the dependencies between tasks and complex arrival situations will be considered.

Qiuyan Liu *et al.*,[12] proposed the problem of how to reduce the average delay of MEC(Mobile-Edge Computing) application by collaborative task scheduling. The collaborative task scheduling is modeled as a constrained shortest path problem over an acyclic graph. By characterizing the optimal solution, the constrained optimization problem is simplified according to one-climb theory and enumeration algorithm. Generally, the **edge-cloud collaborative task scheduling scheme performance better** than independent scheme in reducing average delay. In heavy workload scenario, high blocking probability and retransmission delay at **MEC is the key factor for average delay.** Hence, more tasks executed on central cloud with abundant resource are the optimal scheme. Otherwise, transmission delay is inevitable compared with execution delay. MEC configured with higher priority and deployed close to terminals obtain more performance gain. Research investigated the problem of how to reduce the average delay of MEC application by collaborative task scheduling. The collaborative task scheduling is modeled as a constrained shortest path problem over a cyclic graph. By characterizing the optimal solution, the constrained optimization problem is simplified according to one-climb theory and enumeration algorithm. Generally, the edge-cloud collaborative task scheduling scheme performs better than independent scheme in reducing average delay. In heavy workload scenario, high blocking probability and retransmission delay at MEC is the key factor for average delay. Hence, more tasks executed on central cloud with abundant resource are the optimal scheme. Otherwise, transmission delay is inevitable compared with execution delay. MEC configured with higher priority and deployed close to terminals obtain more performance gain. For **future work researcher consider various extensions of this work.** First, more mathematical models of practical details in cloud computing are necessary to be considered in edge-cloud collaborative execution. Second, power saving on MEC and cloud platform is also important. In addition, the task workflow topology can be extended into more general modes.

Jinke Ren *et al*.,[13] proposed to transform the original joint communication and computation resource allocation problem into an equivalent convex optimization problem and obtain the closed-form computation resource allocation strategy by leveraging the convex optimization theory. Moreover, a necessary condition is also developed to judge whether a task should be processed at the corresponding edge node only, without offloading to the cloud server. Finally, simulation results confirm our theoretical analysis and demonstrate that the proposed collaborative cloud and edge computing scheme can evidently achieve a better delay performance than the conventional schemes. Research proposed to joint communication and computation resource allocation to minimize the weighted-sum delay of all devices in a cloud-edge collaboration system. A latency minimization problem with the constraints of communication and computation resources has been firstly formulated. To tackle it, research has decomposed the optimization problem into two sub problems. The first one is associated with the communication resource allocation whose optimal solution can be expressed in closed-form by using the Cauchy-Buniakowsky- Schwarz inequality. The other sub problem corresponds to the computation resource allocation between the edge nodes and the cloud server. Research has found that the optimal task splitting strategy for each mobile device is determined by two significative parameters: the normalized backhaul communication capacity and the ***normalized cloud computation capacity*.** Researcher has further highlighted some inherent insights of the optimal task splitting strategy by analyzing four special scenarios. Based on this, the optimal computation resource allocation can be finally derived by utilizing the KKT conditions. Our initial study here sheds light on the design of collaborative cloud and edge computing in future mobile computing systems. This research proposed to utilize the weighted-sum delay of all devices as the performance metric, which may not guarantee the latency requirement of each single user. Our analysis can be extended to more general scenarios where each user's latency requirement is considered as an additional constraint in $P1$. **Moreover, to gain closed-form and insightful results, user mobility, asynchronous task arrival,** base station cooperation and server scheduling queuing **are not investigated in this work.** These issues, however, are important and interesting, which can be emphasized in the future work. Also, in our computation model, computing at local devices is not considered. Future works can collaborate local, edge, and cloud computation capacities for further performance enhancement.

Yutao Huang *et al*.,[14] researcher proposed the task scheduling problem for reducing weighted transmission time which takes learning accuracy into consideration. Research proposed was efficient scheduling algorithms which are able to achieve up to 50% reduction in makespan with extensive trace-driven simulations. This research concentrated on the issue of deep learning applications in a cloud-edge learning system. Deep learning with the cloud requires large amounts of training data to support while it brings huge traffic to the network and cost a long time to transmit the data. Edge learning becomes a promising method to solve the transmission issue while brings accuracy reduction. Research proposed scheduling problem for reducing the maximum weighted uploading time on the edge server which also takes accuracy into consideration. Since there is a tradeoff between learning accuracy and traffic reduction, research use the best space remaining ratio for edge learning tasks and reduce the problem to an ILP problem. This research proposed two algorithms and present their efficiency by simulation. This simulation results show that those algorithms are able to outperform the baseline algorithm.

Guang Peng *et al.*,[15] proposed evolutionary large-scale sparse multi-objective optimization (ELSMO) algorithms for collaboratively solving edge-cloud computation offloading problems. To begin with, a collaborative edge-cloud computation offloading multi-objective optimization model is established in a mobile environment, where the offloading decision is represented as a binary encoding. Considering the large-scale and sparsity property of the computation offloading model, the Restricted Boltzmann Machine **(RBM)** is applied to reduce the dimensionality and learn the Pareto-optimal subspace. In addition, the contribution score of each decision variable is assumed to generate new offsprings. Combining the RBM and the contribution score, two evolutionary algorithms using non-dominated sorting and crowding distance methods are designed, respectively. The proposed algorithms are compared with other state-of-the-art algorithms and offloading strategies on a number of test problems with different scales. The experiment results demonstrate the superiority of the proposed algorithms.

This research proposed two evolutionary large-scale sparse multi-objective optimization (ELSMO) algorithms are proposed and compared for solving heterogeneous edge-cloud computation offloading problems. Considering the large-scale and sparsity properties of the multi-objective offloading model, the RBM is used to reduce the dimensionality and learn from the Pareto optimal subspace. While the contribution score is applied to select better decision variables to generate offspring. The proposed algorithms are compared with other MOEAs and offloading schemes to solve the test problems under different scales. The experimental results demonstrate the effectiveness and efficiency of the proposed algorithms. **In the future, some other efficient methods that can reduce the dimensionality will be considered,** such as Principal Component Analysis (PCA) and Autoencoder (AE). And also, the relationship between different decision variables will be investigated.

Shudong Wang *et al.*,[16] proposed to schedule tasks on edge devices or cloud and present a task scheduling algorithm for tasks that need to be transferred to the cloud based on the Catastrophic Genetic Algorithm **(CGA)** to achieve global optimum. The algorithm quantifies the total task completion time and the penalty factor as a fitness function. By improving the roulette selection strategy, optimizing mutation (change) and crossover operator, and introducing cataclysm strategy, the search scope is expanded. Furthermore, the premature problem of the evolutionary algorithm is effectively alleviated. The experimental results show that the algorithm can address the optimal local issue while significantly shortening the task completion time on the basis of satisfying tasks delays. In this research proposed a task scheduling strategy under deadline constraint, where tasks on edge devices could select the execution place including cloud and local devices. And the goal is to minimize the execution time of all tasks. The CGA algorithm as an alternative method to solve the task scheduling problem; this algorithm adds cataclysm strategy to it. Research have considered the constraint of time and optimized the task scheduling. The algorithm CGA was inspired by the behavior of the extinction in the Ice Age, and it is used as a global optimization algorithm. The CGA algorithm research proposed was simulated in the CloudSim environment, and the main objective was to minimize the execution time and meet delay. +e results are compared with the results of existing heuristic methods such as the traditional genetic algorithm (GA) and the time based differential evolution algorithm (TDE). From the experimental results, research also get the conclusion that the proposed CGA can efficiently schedule the tasks to the VM and achieve our goals. In the future, research will consider improving the algorithm under conditions that are closer to the actual environment so that the algorithm can be applied to dynamic and real-time task scheduling in edge-cloud collaboration. Besides, **researcher wants to build a multi-objective version of CGA** for optimizing the task scheduling problem in the cloud. **Study of workflow scheduling using CGA is another future investigation.** And research can also mine or forecast its potential relationships. In addition, the method of task scheduling can consider many other parameters, such as the use of memory, peak of the demand, and overloads. Besides, research can combine the Markov chain with the parallel computing framework and apply it in our model.

Zeng Zeng *et al.*,[17] used the edge resources and minimize the response delay of intelligent applications, research proposed an adaptive task scheduling in a cloud-edge system. First, research proposed coarse-grained and fine-grained services deployment algorithms to solve where and how to deploy, respectively. Then, research model the intelligent tasks as the directed **acyclic graph (DAG) and propose** an adaptive task scheduling algorithm based on the deployment of the services. Finally, these works conduct a large number of simulation experiments and analyze the performance of the algorithms from three aspects: response time, task execution success rate, and QoS. The results show that those algorithms have better performance than the comparison algorithms. In this research abstracted the intelligent application and cloud-edge system, and then formulated the QoS from the users' perspective. Next, in his research proposed the model deployment and the task scheduling strategy separately. For the deployment of the model, research proposed a coarse-grained model deployment algorithm based on location information and a fine-grained model deployment algorithm based on tagging. For the task scheduling strategy, research first constructed a task DAG according to the dependencies between tasks. Research proposed an adaptive task scheduling algorithm for edge intelligent applications in high concurrency scenarios based on the priority of the task and the Earliest Start Time (EST) and Earliest End Time (EFT) of each task. Simulation experiments verify the effectiveness of coarse/fine-grained service deployment and task scheduling that research proposed, respectively. Finally, research combined the deployment and the scheduling strategy for a comprehensive experiment. The results verified that the comprehensive experiment has better results in three aspects: the execution time, the success rate, and the QoS.

Xin Song *et al.*,[18] proposed a cloud-edge collaborative computation offloading schemes based on game theory and prove the existence of Nash Equilibrium. The simulation results demonstrate that proposed algorithm can improve output performance as comparing with the conventional algorithms, and its performance is close to the of the enumerative algorithm. In this research proposed a cloud-edge collaborative computing task scheduling and resource allocation algorithm, which minimizes the total system cost. The system model including cloud center, edge server and the sensor nodes is built for the cloud-edge integrated **EI** network, where each sensor node has an indivisible task that can be executed locally, at the edge node, or at a remote cloud cooperatively. To improve the efficiency of EI and the QoS of the energy applicants, research proposed a joint task offloading and resource allocation scheme under the limited communication and computation resource constrains, in which the optimization problem is a NP-hard problem, which is difficult to solve. To obtain the optimal solution, the optimization problem is divided into the power allocation sub-problem, the computation resource allocation sub-problem and the offloading scheme selection sub-problem. **A bisection search algorithm is developed to obtain the optimal power allocation for each sensor node.** Then, experiments derive the optimal computation resource allocation of the edge server by KKT condition and convex optimization theory. Furthermore, research will establish a game model to obtain the optimal task offloading scheme. The simulation results show that the proposed algorithm can significantly reduce the total system cost, has a fast convergence rate, and decrease the communication and computation delay compared with conventional approaches.

Xiaolong Xu *et al.*,[19] proposed a vehicle-to vehicle communication-based route obtaining algorithm is designed first. Then, in order to ensure the trust worth of the IoV data, which ECD to upload the computing tasks to is selected. Under the case that all ECDs are overloaded, the computation offloading between ECDs and cloud is considered. In addition, non-dominated sorting genetic algorithm III is adopted to realize the multi-objective optimization of decreasing the load balancing rate and reduce the energy consumption in ECDs and shorten the time during processing the computing tasks. Furthermore, research employs the simple additive weighting and multiple criteria decision making to evaluate the solutions of this proposed method. Finally, experimental evaluations are conducted to validate the efficiency and effectiveness of proposed method.**MOC is proposed in this research**. First, the computation offloading problem is analyzed first with the three objections formulized. Then, NSGA-III is utilized to achieve the multi objective optimization. In addition, a solution evaluation algorithm with SAW and MCDM is designed.

Subsequent experimental evaluations prove the efficiency and effectiveness of MOC. **In future work, research will attempt to reduce the transmission time** of computing tasks during vehicles to reduce the total time consumption of computation offloading while taking load balancing into consideration, and adapt and extend the proposed method to a real-world scenario of IoV services.

Zeyuan Yang *et al.*,[20] research investigated the heterogeneous task scheduling for ECSs over multilayer elastic optical network (MLEON), which involves the joint allocation of heterogeneous computing resources in edge and cloud servers and high-dimensional network resources. Research proposed a Task-Node Matching Score (TNMS) based method, which evaluates the fitness for each mapping tuple between each task in ECS and each substrate node in ML-EON, and adaptively generates a specific matching score for each task-node pair. Furthermore, TNMS is extended with a pre-allocation mechanism (TNMS-Pre) to estimate the costs of multi dimensional resources in ML-EON for virtual link (VL) mapping. The estimated VL mapping costs are integrated into the matching scores to guide the task placement to be cost-efficient. To guarantee the feasibility, a maximal weight matching **(MWM)** based method is presented to determine the task placement schemes. Simulation results demonstrate the effectiveness of the adaptive scoring for heterogeneous task placement and the preallocation mechanism for reducing the ML-EON costs.
This proposed the heterogeneous task placement in the edge-cloud collaboration scenario over ML-EON. TNMS is proposed to adaptively evaluate the task placement scheme for each specific ECS request. To further reduce the ECS cost of VL mapping in ML-EON, TNMS Pre is advocated with a pre-allocation mechanism to integrate the cost of VL mapping into the task placement process. **The proposed methods outperform benchmarks in blocking ratio, ECS capacity, and the average cost** of ECSs. The computational efficiencies of the proposed methods are also demonstrated.

Danlami Gabi *et al.*,[21] proposed a Multi-Objective QoS model to address customers profit based on execution time and execution cost criteria is presented. A Cloud Scalable Multi-Objective Cat Swarm Optimization (CSM-CSO) based Simulated Annealing (SA) (CSM-CSOSA) algorithm is then proposed to solve the model. In this method, the Taguchi Orthogonal approach is used to enhanced the SA and incorporated into the local search of the proposed algorithm for enhancing it exploration capability. Implementation of the algorithm is carried out on CloudSim tool and evaluated using one dataset (Normal distributed) and one Parallel Workload (High-Performance Computing Center North (HPC2N)). Quantitative analysis of the algorithm performance is taken based on metrics of execution time, execution cost, QoS and percentage improvement. Result obtained is compared with that of Multi Objective Genetic Algorithm (MOGA), Multi-Objective Ant Colony (MOSACO) and Multi Objective Particle Swarm Optimization (MOPSO), where proposed method is able to returned substantial performance with improved QoS. **This Research work proposed multi-objective task scheduling problem that accounts for consumers' QoS** expectations are formulated and a multi-objective scheduling model in relation to the scheduling problem was proposed based on execution time and execution cost criteria. A CSM-CSOSA task scheduling optimization algorithm is then proposed that solved the multi-objective scheduling model. The novelty of the proposed method was based on the use of SA and Taguchi experimental design procedure that enhanced the local search procedure of the algorithm in exploring larger search space that returned better optimum solutions. Comparison of the performance of CSM-CSOSA with some of the existing metaheuristics (MOPSO, MOSACO, MOGA) task scheduling algorithms is carried out with one dataset and one parallel workload. The results obtained shows proposed method achieved better performance by returning good QoS.

Chunlin Li *et al.*,[22] proposed adaptive resource allocation algorithm and a data migration algorithm. The prediction algorithm provides the basis for the adaptive resource allocation of the edge cloud cluster. The adaptive resource allocation algorithm determines the resource allocation scheme of the edge cloud cluster with the lowest service cost. The data migration algorithm guarantees the reliability of data and achieves cluster load balancing. A large number of experimental results show that our newly proposed algorithm can greatly improve system performance in terms of better cost control, higher data integrity and load balancing. Research presented a design a load prediction model based on ARIMA and BP neural network. By combining the prediction methods, the prediction results are more accurate which provides a basis for the resource allocation in advance. When adaptive resource allocation is implemented, an adaptive resource allocation algorithm based on billing granularity is designed by considering the heterogeneity of clusters. It allocates resources to minimize costs. Due to the problem of unbalanced cluster load caused by resource expansion and the data reliability caused by resource shrinkage**, the automatic resource placement algorithm is used to implement data migration.** The algorithm can implement load balancing of the edge cloud after resource expansion, and ensure data reliability after resource shrinkage.

Roxana-Gabriela Stan *et al.*,[23] researcher build a scheduling and evaluation framework and measure typical scheduling metrics such as mean waiting time, mean turnaround time, makespan, throughput on the Round-Robin, Shortest Job First, Min-Min and Max-Min scheduling schemes. Our analysis and results show that the state-of-the-art independent task scheduling algorithms suffer from performance degradation in terms of significant task failures and non optimal resource utilization of datacenters in heterogeneous edge–cloud mediums in comparison to cloud-only mediums. In particular, **for large sets of tasks, due to low battery or limited memory, more than 25% of tasks fail** to execute for each scheduling scheme
The experimental outcomes revealed the acute (keen) need of novel scheduling methods to address the challenges imposed by the heterogeneity of tasks and resources. The existing schedulers generate failures by assigning resources with inherent capacity limitations to tasks with strict processing requirements. For reasonably large execution workloads, edge devices run out of battery. The conventional scheduling schemes do not properly tackle the higher battery consumption of computationally intensive tasks. The traditional techniques are not energy-aware, without focusing on the power consumption of the computing machines. The scheduling policies should consider the tasks' arrival as an extra parameter, given the experienced poor performance results in our realistic edge–cloud experiments. **As discussed, the Round-Robin and SJF routines allocating heterogeneous resources to tasks suffer** dramatically from performance degradation.

Leilei Zhu *et al.*,[24] established a layered excellent gene replication strategy (HEGPSO model), in which the optimal replicator is added, and an evolutionary particle swarm optimization algorithm is proposed. In each iteration, the population is divided into three layers based on individual fitness. After that, the optimal replication factor is added to each layer of individuals to enhance the global search ability of the algorithm and ensure the good convergence of the algorithm. **Finally, a balanced resource allocation model is established.** Experiments show that the HEGPSO model proposed in this research has high fitness and fast convergence speed, and is suitable for large-scale task access scenarios. Research proposed an adaptive HEGPSO model based on the characteristics of concurrent task requests in edge cloud. Based on the elite preservation strategy, the model replicates some information of the superior individuals of the previous generation in different degrees according to the superior level of particles. Combined with BPSO algorithm, the concept of the superior replication operator is determined by mode theory. Considering the real-time response requirements of edge environment, the algorithm is compared with the comparison algorithm from two aspects of load balance fitness and convergence speed. By designing equal task request space and increasing task request space, the actual task request scene is simulated. Experimental results show that the proposed algorithm has some advantages in optimizing edge cluster load balancing. **The model will be further improved by adding location influencing factors in the future.**

Hengliang Tang *et al.*,[25]research proposed the dynamic resource allocation algorithm for cloud-edge environment. The dynamic resource allocation algorithm consists of the resource scheduling algorithm and the resource matching algorithm. In the resource scheduling algorithm, a resource scheduling problem can be obtained according to the stored penalty of scheduling contents, the value of scheduling contents and the transmission cost of scheduling contents. Then, tabu search algorithm is applied to find the optimal solution to the resource scheduling problem. Furthermore, the resources are scheduled into the edge servers from cloud datacenter with the optimal solution. In the resource matching algorithm, an optimization problem of the resource matching is built with respect to the resource location, the task priorities and the network transmission cost. **For addressing this problem, the optimal problem is converted to an optimal matching problem** of the weighted bipartite graph. Moreover, an optimal matching problem of the weighted complete bipartite graph is created by adding the spurious containers. Then, the optimal strategy of the resource matching for tasks on the edge servers is achieved. Finally, the performance of the proposed algorithms and some typical resource allocation algorithms is evaluated via extensive experiments. The results indicate that proposed algorithms can effectively reduce network delay and enhance QoS. In this research proposed the dynamic resource allocation algorithm for the cloud-edge environment. The dynamic resource allocation algorithm consists of the resource scheduling algorithm from cloud to edge and the resource matching algorithm on edge servers. The resource scheduling algorithm from cloud to edge is applied to improve the performance of the resource matching algorithm. In the resource scheduling strategy from cloud to edge, a **resource scheduling problem can be obtained according to the stored penalty of scheduling contents,** the value of scheduling contents and the transmission cost of scheduling contents. The stored penalty of a scheduling content is determined by the stored space of ES and the size of scheduling content. When stored space of ES is enough for the scheduling data, there exists no stored penalty. Otherwise, ES needs to release some contents to make room for storing the scheduling content. In this research proposed the dynamic resource allocation algorithm is proposed based on the stably stored capacities of the edge servers for duration of time. In an actual environment, the stored capacities change dynamically. Therefore, the dynamic resource allocation strategy combining with the dynamically stored capacities of the edge servers is a promising future research direction. **Due to the limitation of the dataset in this experiment, the experiment with larger scale dataset** is also a significant extension of our work.

Chunlin Li *et al.*,[26] proposed to combine the optimal placement of data blocks and the optimal scheduling of tasks to reduce the computation delay and response time for the submitted tasks and improve user experience in edge computing. In optimal placement of data blocks, the value of the data blocks considers not only the popularity of the data blocks, but the data storage capacity and replacement ratios of an edge server that will store those data blocks. Furthermore, the replacement cost for placed data blocks is regarded as an important component of data block placement. **This optimal placement scheme can avoid replacing the placed data blocks repeatedly** so that the bandwidth overhead is reduced. In optimal scheduling of tasks, the containers are taken as the lightweight resource unit for the services for user requests to make full use of data storage in edge servers and improve the services performance of edge servers. Finally, extensive experiments are conducted to value the performance of task scheduling strategy. The results show that the performance of the proposed task scheduling algorithm is better that that of the compared algorithms. This proposed work the main purpose is to combine the optimal placement of data blocks and the optimal scheduling of tasks to reduce the computation delay and response time for the submitted tasks and improve user experience in edge computing. In optimal placement of data blocks, the value of data blocks considers not only the popularity of the data blocks, but the data storage capacity and replacement ratios of an edge server that will store those data blocks. Furthermore, the replacement cost for data blocks is regarded as an important component of data block placement. **Finally, extensive experiments are conducted to value the performance of task scheduling strategy.** The results show that the performance of the proposed task scheduling algorithm is better that that of the compared algorithms in terms of HRT and RTT. In the future works, it is worth studying the distribution of edge servers in edge computing. As a novel paradigm, **how to deploy edge servers for better user experience is a significant problem**

III.CONCLUSION

Task scheduling plays a critical role in the performance of the edge-cloud collaborative. Whether the task is executed in the cloud and how it is scheduled in the cloud is an important issue. On the basis of satisfying the delay, ***various researches was scheduled tasks on edge devices or cloud*** and present a task scheduling algorithm for tasks that need to be transferred to the cloud based on the Optimization algorithm to achieve global optimum. Tasks can be scheduled to the edge or the far (remote) cloud based on energy consumption and time delay. For the problem that needs to be processed in the cloud center, ***how to perform proper task scheduling to achieve the goal is worthwhile (valuable) research question.*** Task scheduling methods in the cloud center can be divided into heuristic algorithms (such as Round Robin, Min-Min, Max-Min and Shortest Job First), metaheuristic algorithms (based on biological incentives and swarm intelligence), and hybrid task scheduling algorithms. In task scheduling process, various performance-based performance indicators such as system utilization, execution time, load balance, network communication cost, delay, and the like are used. The heuristic task scheduling algorithm can easily schedule tasks and provide the best solution. However, it does not guarantee the best results and is easy to fall into partial selection. The metaheuristic algorithm is an improved algorithm based on a heuristic algorithm, which is a combination of random algorithms and local search algorithm. It enables the exploration and development of search space and handles a large amount of search space information. In addition, it can use learning strategies to acquire and master information to effectively find approximate optimal solutions. Among them, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Algorithm (ACO) are the most widely used evolutionary algorithms in the task scheduling in recent years. ***However, these algorithms usually converge prematurely and are prone to finite optimally.*** When approaching the optimal solution, it may also swing left and right, making the convergence slower. **In future, our research plan is to be implement task scheduling** using infinite optimal algorithms such as Bacterial Colony Optimization **(BCO),** Social Spider Optimization **(SSO),** Lion Optimization Algorithm and Elephant Herding Optimization Algorithm.

IV.REFERENCES

[1]. Himani K Langhnoja, Prof Hetal A JoshiyaraMulti-Objective Based Integrated Task Scheduling In Cloud Computing, IEEE Conference Record # 45616; IEEE Xplore ISBN: 978-1 7281-0167-5, Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019]

[2]. Chang H, Hari A, Mukherjee S, et al. Bringing the cloud to the edge. Computer Communications Workshops. IEEE, 2014: 346-351

[3]. J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, ``A survey on end-edge cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," ACM Comput. Surv., vol. 52, no. 6, Oct. 2019.

[4]. M. A. Elaziz, S. Xiong, K. P. N. Jayasena, and L. Li, “Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution,” Knowledge-Based Systems, vol. 169, pp. 39–52, 2019.

[5]. J. Xue, L. Li, S. Zhao, and L. Jiao, “A study of task scheduling based on differential evolution algorithm in cloud computing,” in Proceedings of the International Conference on Computational Intelligence Communication Networks, Bhopal, India, November 2014.

[6]. Z. Xu, W. Liang, M. Jia, M. Huang, and G. Mao, ``Task offloading with network function requirements in a mobile edge-cloud network," IEEE Trans. Mobile Comput., vol. 18, no. 11, pp. 2672_2685, Nov. 2019.

[7]. Firdose Saeik, Marios Avgeris, Dimitrios Spatharakis, Nina Santi, Dimitrios Dechouniotis,”Task Offloading in Edge and Cloud Computing: A Survey on Mathematical, Artificial Intelligence and Control Theory Solutions”, 2021

[8]. XueLi Yao,”A Multi-Objective Cloud Workflow Scheduling Optimization Based on Evolutionary Multi-objective Algorithm with Decomposition”, Journal of research square, https://doi.org/10.21203/rs.3.rs-604125/v1

[9]. Asad Mahmood, Yue Hong, Muhammad Khurram Ehsan and Shahid Mumtaz,”Optimal Resource Allocation and Task Segmentation in IoT Enabled Mobile Edge Cloud”,

[10]. Mahbuba Afrin, Jiong Jin, Ashfaqur Rahman, Yu-Chu Tian, Ambarish Kulkarni,”Multi-Objective Resource Allocation for Edge Cloud based Robotic Workflow in Smart Factory”, 2019

[11]. Zheng Shi, Zhiguo Shi,”Multi-node Task Scheduling Algorithm for Edge Computing Based on Multi-Objective Optimization”, Journal of Physics: Conference Series,2020, doi:10.1088/1742-6596/1607/1/0120171

[12]. Qiuyan Liu, Jiajun Li, Huazhang Lv, Zhonghao Zhang, Mingxuan Li, Yi Feng,”Edge-Cloud Collaborative Optimization Scheduling with Micro-Service Architecture”, Journal of Computer and Communications, 2019, 7, 94-104, ISSN Online: 2327-5227

[13]. Jinke Ren, Guanding Yu,”Collaborative Cloud and Edge Computing for Latency Minimization”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY,, 2018

[14]. Yutao Huang, Yifei Zhu, Xiaoyi Fan, Xiaoqiang Ma, Fangxin Wang, Jiangchuan Liu,”Task Scheduling with Optimized Transmission Time in Collaborative Cloud-Edge Learning”,

[15]. Guang Peng, Huaming Wu, Han Wu and Katinka Wolter,”Evolutionary Large-scale

Sparse Multi-objective Optimization for Collaborative Edge-cloud Computation Offloading", ISBN: 978-989-758-475-6, In Proceedings of the 12th International Joint Conference on Computational Intelligence (IJCCI 2020), pages 100-111

[16]. Shudong Wang , Yanqing Li , Shanchen Pang , Qinghua Lu, Shuyu Wang, and Jianli Zhao3,"A Task Scheduling Strategy in Edge-Cloud Collaborative Scenario Based on Deadline", Volume 2020,journal hindawi

[17]. Zeng Zeng, Weiwei Miao, Shihao Li, Xiaoyun Liao, Mingxuan Zhang, Rui Zhang, for Edge Intelligence Application", 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

[18]. Xin Song, Yue Wang, Zhigang Xie, Lin Xia ," A Cloud-Edge Collaborative Computing Task Scheduling and Resource Allocation Algorithm for Energy Internet Environment", KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 15, NO. 6, Jun. 2021

[19]. Xiaolong Xu,Renhao Gu, Fei Dai, Lianyong Qi, Shaohua Wan,"Multi-objective computation offloading for Internet of Vehicles in cloud-edge computing", Wireless Networks,2019

[20]. Zeyuan Yang, Rentao Gu, Zuqing Zhuy, Yuefeng Ji,"Edge-cloud Collaborative Heterogeneous Task Scheduling in Multilayer Elastic Optical Networks"

[21]. Danlami Gabi, Abdul Samad IsmailCloud Scalable Multi-Objective Task Scheduling Algorithm for Cloud Computing Using Cat Swarm Optimization and Simulated Annealing, 2017 8th International Conference on Information Technology (ICIT)

[22]. Chunlin Li, Hezhi Sun , Hengliang Tang , Youlong Luo,Adaptive Resource Allocation Based on the Billing Granularity in Edge-Cloud Architecture, Computer Communications, 2019

[23]. Roxana-Gabriela Stan , Lidia B˘ajenaru, C˘at˘alin Negru and Florin Pop,"Evaluation of Task Scheduling Algorithms in Heterogeneous Computing Environments", 2021

[24]. Leilei Zhu, Ke Zhao, Huaze Lin ,Dan Liu, and Li Li,"Edge cloud task scheduling model based on layered excellent gene replication strategy", Journal of Physics: Conference Series, 2132 (2021) 012002

[25]. Hengliang Tang,"Dynamic resource allocation strategy for latency-critical and computation intensive applications in cloud-edge environment", Computer Communications, 2018

[26]. Chunlin Li, Jingpan Bai, JiangHang Tang,"Joint optimization of data placement and scheduling for improving user experience in edge computing", J. Parallel Distribution of Computing, 2018

ABOUT THE AUTHORS

K.Vinothkumar received his **M.Phil** Degree from Periyar University, Salem in the year 2015. He has received his **M.Sc.,** Degree from Hindusthan College of Arts and Science, Coimbatore affiliated to Bharathiar University, Coimbatore in 2013. He is pursuing his **Ph.D** Degree (Part-Time) in Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India. He is working as **HOD Cum Assistant Professor** in Department of Computer Science at Kavitha's College of Arts and Science (Co-Ed.,), Vaiyappamalai, Tiruchengode, Namakkal. His current research of interests includes Edge Computing, Cloud Computing, Mobile Computing and Ad hoc Networks.

Dr.D.Maruthanayagam received his **Ph.D** Degree from Manonmaniam Sundaranar University, Tirunelveli in the year 2014. He received his **M.Phil** Degree from Bharathidasan University, Trichy in the year 2005. He received his **M.C.A** Degree from Madras University, Chennai in the year 2000. He is working as **HOD Cum Professor**, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India. He has above **21 years** of experience in academic field. He has published **7 books**, more than **55 papers** in International Journals and **30 papers** in National & International Conferences so far. His areas of interest include Computer Networks, Grid Computing, Cloud Computing and Mobile Computing.