Task-Driven Resource Assignment in Mobile-Edge Computing Exploiting Evolutionary Computation

Article *in* IEEE Wireless Communications · September 2019

DOI: 10.1109/MWC.001.1800582

Task-Driven Resource Assignment in Mobile-Edge Computing Exploiting Evolutionary Computation

Liangtian Wan, Lu Sun, Xiangjie Kong*, Yuyuan Yuan, Ke Sun, and Feng Xia
Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software,
Dalian University of Technology, Dalian 116620, China

***Abstract*—**The Internet of Things (IoT) network allows IoT devices to communicate with other devices, applications and services by exploiting existing network infrastructure. Recently, a promising paradigm, mobile-edge computing (MEC), emerging for alleviating high latency data services in cloud computing framework plays an important role in the IOT network. The network performance and intelligence can be improved by integrating cognitive and cooperative mechanisms in MEC framework. However, the quality of service (QoS) of computational intensive tasks may degrade because of the limited available computational resources in MEC servers. Moreover, the characteristics of resources belong to MEC servers and cloud servers are commonly different. In order to optimize the strategy of resource assignment, the tasks of assigning the limited computational resources in MEC servers and resolving the high latency problem in cloud servers have attracted growing interests from researchers. In this paper, we propose a joint optimization paradigm for task-driven resource assignment based on evolutionary computation considering both the power consumption and computation/communication delay simultaneously. MEC framework consists of MEC servers, mobile devices and cloud servers, and offloads the computational resources to the edge of end-users. Additionally, we introduce and analyze three typical task-driven based cases, which are server-determined condition, server-flexible condition, and server-uncertain condition, respectively. Finally, we give the existing technical challenges and discuss the open research issues.

I. Introduction

In recent years, the number of mobile terminals has grown explosively, such as smartphones, wearable devices and vehicle terminals. The Internet of Things (IoT) network integrating existing network with cognitive and cooperative mechanisms has been developed greatly based on the rapid commercial application of these mobile terminals. The existing cloud computing framework can support numerous PC end-users, but cannot provide the diverse services for the increasing number of smart mobile devices. From the prediction of Cisco, the average number of mobile devices per person will reach 6.58 in 2020 [1]. Meanwhile, people prefer high-quality mobile data services, thus providing satisfied quality of service (QoS) to mobile users becomes a critical issue. The term “Fog computing” is extended from the concept of cloud computing. Fog computing, also known as mobile edge computing (MEC), offloads data into the MEC servers, which are close to the end-users, instead of almost delivering all data and tasks into the cloud. MEC servers, such as commercial edge routers, set-top-boxes and access points, provide service resources at the edge of network [2]. There are several fundamental characteristics of MEC such as low latency, location-awareness, geo-distribution and mobility support that make MEC scalable to extend the computing ability from the center to end user at the edge of networks by providing elastic and deployable resources. The discrimination between cloud computing and fog computing can be elaborated in a vivid way as follows. MEC (fog computing) is closer to the ground than the cloud, and the data for various services is generated on the ground. The data is closer to MEC compared with cloud computing. The data and computing power in edge computing is moving from data center to the edges which are close to users. Compared with edge computing, mobile edge computing focuses on the user experience of mobile users in cellular networks.

MEC performs as the intermediate layer between the cloud and mobile users, and it is a distributed computing framework as well. The computations, communications, control and resources services are distributed near to end-users or network devices, which are close to the end-users. Consequently, users’ requests can be processed in real-time and the deficiencies of the cloud computing framework can be alleviated dramatically.

A large number of MEC servers constitute the MEC platform. These MEC servers can be scattered in different geographic locations, and they are opposite with the data centers whose resources are concentrated. However, it is not easy to allocate resources reasonably and provide high QoS for mobile users, especially in the scenarios of the mobile devices with large scale. In recent years, for the sake of increasing networks scalability and reducing power consumption, some technologies for relieving the pressure from implementation and management of resources have been proposed, such as software-defined networking (SDN) and network function virtualization (NFV).

Resource assignment is extremely challenging because of the restrictions on bandwidth, energy, storage, computational resources of mobile devices. Thus, the coordination among MEC servers and cloud servers is required in order to optimize the deployment of resources as well as to minimize the consumption delay.

As an essential part of cognitive computing, evolutionary computation has attracted much attention in the field of resource management and industrial scheduling, etc. Evolutionary algorithm (EA) is a random search technique which is based on natural selection and natural genetics. The core idea of EA is to maintain a balance between exploitation

*Corresponding author: Xiangjie Kong; email: xjkong@ieee.org

and exploration in order to find the optimal solution to survive in a variety of environments. In the complex solution space, EA has strong robustness and searching ability. The existing researches prove that EA performs well in combinational optimization problems, especially for the applications of resource allocation. *In this paper, a task-driven resource assignment paradigm based on evolutionary computation is proposed for MEC. In this framework, the MEC servers and cloud servers are both considered, since the QoS of the computational intensive tasks may degrade due to the limited computational resource in MEC servers. Therefore, these tasks offloading to the MEC servers and cloud servers should be scheduled jointly. By taking the server condition into consideration, we take three case studies to understand the resource assignment in MEC system, i.e., server-determined condition, server-flexible condition and server-uncertain condition. The methods of evolutionary computation have been utilized for resource assignment in each condition. We have verified the effectiveness of the evolutionary computation methods based on synthetic dataset and real world dataset in MEC framework.*

The rest of the paper is organized as follows. We first analyze and discuss state-of-the-art resource assignment techniques for MEC (fog computing) from various viewpoints in Section II. Then the task-driven MEC framework based on evolutionary computation is proposed and analyzed in Section III. The three case studies, analyzing the serviceability and advantage of task-driven resource assignment in MEC framework, are described in the sequent three sections. The main technical challenges and open research issues that should be resolved are discussed in Section VII. Finally, we conclude the paper in the last section.

II. State of the Art

Due to the resource restriction, resource heterogeneity and dynamic nature of resource requirement, the resource management in MEC environment is difficult to tackle. Resources assignment contributes to utilize the restricted resources efficiently and improves the QoS of users, etc. Computational requirements offloading to cloud computing systems have attracted much attention from researchers, while there are few reports about the works of resource assignment in MEC. Next, we analyze the state-of-the-art resource assignment methods proposed and applied in the literatures.

Table 1. Comparisons of state-of-the-art methods for MEC.

	Hardware	Advantages	Computation complexity	Scalability	Application delay
Zeng et al. [3]	Very small, the storage servers with fixed computational resources.	The network edges are equipped with moderate storage.	Low.	High.	Long queuing delay.
Gu et al. [4]	Highly capable computing base stations.	Integrate MCPS with fog computation to construct fog computing supported MCPS.	Low.	High.	Delay are not strict since all resources are sufficient.
Deng et al. [5]	Depends on the allocated workload.	The low service delay because of the low resource consumption.	High.	Low.	The trade-off between the service delay and power consumption.
Do et al.[6]	Small, heterogeneous devices with moderate computing resources.	Decompose the large problem into many sub problems.	Low.	High.	The latency depends on the geographical distribution of end-users.
You et al. [7]	Minimize mobile energy consumption.	Low-complexity.	Low.	High.	Short latency.
Liu et al. [8]	Mobile devices and MEC servers with moderate computing resources.	The one-dimensional algorithm with low power-constrained delay.	Low.	High.	A shorter average execution delay.
Tran et al. [9]	Applications with moderate computing resources.	Collaborative MEC paradigm.	Low.	High.	Short latency.
Plachy et al. [10]	Base stations with moderate computation resources.	Flexible selection of communication path together with VM placement.	Low.	High.	Short latency.
Wang et al. [11]	MEC servers with moderate computation resources.	Service migration using the framework of Markov Decision Process (MDP).	Low.	High.	Short latency.
Kosta et al. [12]	Dynamically request VMs with more computational power.	Thinkair framework, migrate smartphone applications to the cloud.	Low.	High.	Parallelizable application reduces latency.

Once the data services break off, Zeng *et al.* [3] suggested that returning the remaining resources to the target user. Their approach for resource assignment minimize the resource consumption. For instance, they would not load the complete image into cloud servers. Instead, multiple parts of image are stored in the servers at the edge, and then users can retrieve the image from these edge servers immediately and conveniently. In order to alleviate the high consumption of computational resources in medical cyber-physical systems (MCPS), which is the combination of MEC and medical cyber-physical systems, Gu *et al*. [4] minimized the overall consumption by exploiting base station association, task distribution and VM deployment while satisfied the QoS of users simultaneously. Deng *et a*l. [5] designed a representative framework for monitoring the communication/consumption delay problem between the fog and cloud computing paradigms, and indicated that the minimal consumption would be achieved by constraining the consumption delay of various services. Furthermore, Do *et al*. [6] considered the carbon footprint and maximized the efficiency of resource utilization simultaneously. They developed a video streaming service with MEC servers and proposed an optimized algorithm with fast convergence.

In addition, You et *al*. [7] studied the optimal resource assignment strategy with low complexity under the scenarios of time division multiple access (TDMA) and orthogonal frequency-division multiple access (OFDMA) based systems. Liu *et* al. [8] utilized the Markov Decision Process (MDP) for optimizing and deploying the task of resource allocation with low delay. A framework residing at the edge of the radio access networks (RAN) is envisioned in [9]. The framework comprises mobile devices and MEC servers, and the overall resources are integrated in the edge servers.

In terms of the service migration problem, the MDP is a widely used framework to determine the migration situations, including the time and location of migration. Plachy *et* al. [10] dealt with user's mobility from two patterns: migrating the application component or mining an appropriate new path for communication between MEC servers and the mobile devices. Wang *et al*. [11] considered the mobility of a user as a two dimensional random walk model, and regarded the migration of services as an MDP based process on the distance.

Kosta *et al.* [12] designed an offloading framework for resource allocation based on the cloud architecture and further simplified the process of migration between mobile devices and cloud. Shojafar *et al.* [13] proposed an energy-efficient adaptive resource management method for maximizing computational efficiency, as well as better satisfying the requirements of QoS. Ni *et al.* [14] designed a resource assignment scheme, which took the performance metrics such as benefit and energy consumption in a task and the creditability evaluation on target users, as well as the fog resources into consideration, but it ignored other evaluation metrics, e.g., the time complexity of their methods, etc.

In summary, prior works focus on the time and price consumption, carbon footprint, energy consumption, etc. Even though they consider resource assignment problems from different perspectives, the objective of all researchers are identical, i.e., to improve application performance and maximize the benefits obtained by both service providers and users. We list the comparisons of their allocation methods from various aspects in Table 1. *Most existing works concern that MEC have not noticed the scalability of resource allocation between MEC and cloud servers. While this paper proposes a task-driven resource assignment paradigm based on evolution computation that considers the power consumption and computation/communication delay, and then introduces three representative case studies for efficient application of our scheme.*

Previous work mainly focus on cloud servers or MEC servers, and they rarely take both cloud and MEC servers together to make the resource allocation, especially for some typical scenarios in the applications of MEC servers. i.e., the tasks in the identical mobile device are ordered as a prior, each task can be processed in one specific MEC or cloud server or multiple MEC and/or cloud servers, and the processing time of each task maybe uncertain. These constraints have not been considered in previous work, and the uncertain processing time is a typical scenario for task execution in MEC or cloud servers.

III. WHAT IS TASK-DRIVEN RESOURCE ASSIGNMENT ?

According to the definitions of International Telecommunication Union (ITU), the key indicators for the three major application scenarios of 5G are 10Gbps peak throughput rate, 1ms delay, 1 million connection number, and 500km/h high speed mobility. However, in the traditional network architecture, the core network is deployed at a far location related to users and the transmission delay is large, which obviously cannot meet the ultra-low latency service requirements. In addition, the transmission of massive data to the cloud servers also wastes bandwidth and increases the delay.

In order to overcome long and unpredicted delay in cloud computing mode, the edge computing has been considered as an emerging paradigm that supports real-time and mobile data services. It has been proved that MEC system is an effective framework for resolving the restrictions of communication/consumption delay in many applications. The MEC server (fog server) such as wireless IP camera, the routers and switches are deployed locally, and thus some tasks can be executed in the MEC servers. However, the QoS of the computational intensive tasks may degrade due to the restricted computational resource in MEC servers. If a large number of tasks are executed in MEC servers, there may exist vacancy in the cloud server. In fact, there are different characteristics between the MEC servers and the cloud servers, i.e., they are expert in different domains. Meanwhile, mobile users have different requirements of QoS. In order to better serve the mobile users, the operation of the computational intensive tasks offloading to the MEC servers and cloud servers should be scheduled jointly, which is called task-driven resource assignment.

An overall architecture of the task-driven MEC framework is illustrated in Fig. 1. The service requirements from the mobile users are received through the interfaces such as keyboard and touch screen. The data transmission has two forms: direct data transmission and indirect data transmission explicitly. The direct data transmission is to offload the tasks into MEC servers or the cloud server directly without intermediate layers.

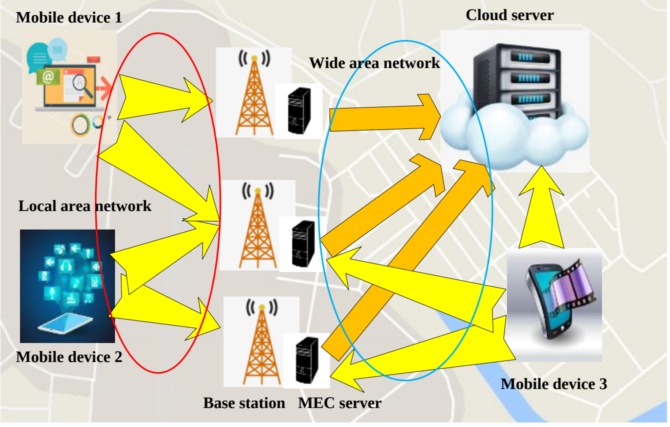


Figure 1: Task-driven MEC framework.

As shown in Fig. 1, the tasks of the mobile devices 1, 2 and 3 can be offloaded to the MEC servers directly without the forwarding to the base station. These requirements are separately input to the MEC servers through a local area network (LAN). The indirect data transmission is to offload the tasks into the cloud servers indirectly with the forwarding to the base station. As shown in Fig. 1, the tasks of the mobile devices 1 and 2 can be offloaded to the cloud server indirectly with the forwarding to the base station. *The unprocessed requirements are dispatched to the cloud server from the MEC server via the wide area network (WAN), which covers a wide area from MEC servers or base station to the cloud server in geography*. The deployment of MEC servers have close proximity to the mobile devices. Compared with WAN, the delay of LAN communication can be omitted. We will consider and analyze the power consumption of MEC servers and cloud servers, the computation/communication delay of MEC, WAN communication and cloud computing for further understanding MEC servers and cloud servers in MEC framework.

The resource allocation is related to location distribution of MEC servers. If one MEC server is closer to the user than another MEC server, then the task offloading of the user's mobile device would delivered to the former MEC server rather than the later MEC server. Thus, the location distribution of MEC servers would affect the resource allocation strategy.

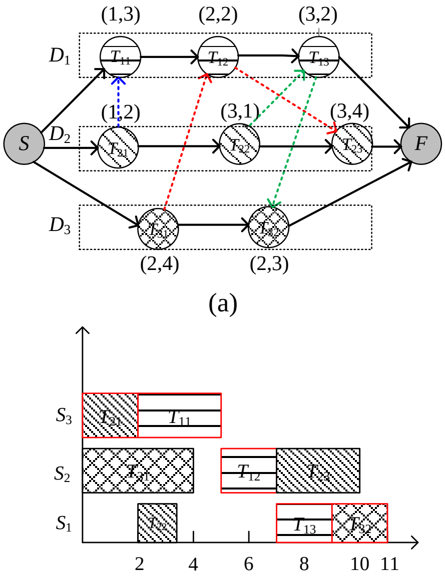
EA is a robust and widely used optimization paradigm. It has the characteristics of self-organization, self-adaptation and self-learning. It enables to solve the complex problem that could not be effectively solved by traditional optimization algorithms, e.g., combinational optimization problems, NP-hard problems, etc. Recently, EAs are widely applied in parameters optimization, industrial scheduling, complex network analysis and resource allocation. In our framework, there exist several mobile devices and each mobile device provides a certain number of tasks, which will be processed on a set of servers in MEC system with a predefined order and processing times (communication delays and computation delays). The tasks in the identical mobile device are ordered as a prior, and each task can be processed in one specific MEC or cloud server or multiple MEC and/or cloud servers. The processing time of each task maybe uncertain. We should formulate the resource assignment in MEC system by taking the resource constraint into consideration. This means that we cannot optimize the resources allocation problem by only considering task order or resource allocation. The resource allocation in MEC framework is a typical combinational optimization problem, and thus we adopt EAs to solve the resource allocation problems in MEC framework for different cases as described in the following sections.

IV. Case Study I: Server-Determined Condition

Although there are restricted resources of mobile devices, some applications using various techniques, such as machine learning, artificial intelligence, and data mining, have to work uninterruptedly with seasonable feedback [15]. Nevertheless, a common issue, the computation amount with large consumption, exists in all the techniques mentioned above. Although cloud servers have strong ability of computing, the delay caused by the data transmission of traditional approaches cannot be acceptable. MEC is leveraged to deploy applications and services in which servers are close to mobile devices. All tasks of mobile devices can be viewed as different small tasks to be processed on each server in MEC system. Since specific resources are needed for some issues while not all servers can satisfy the requirement, some servers have to be determined for certain task, e.g., for image processing, the servers equipped with high-performance graphic processing units (GPUs) are preferred.

In this case, we propose a server-determined MEC framework. The tasks of mobile devices are assigned to the specific servers enabling fixed resources. Various EAs can be adopted for this framework, e.g., particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), etc. Different EAs have different characteristics, and different encoding mechanisms affect the performance and robustness as well. Besides these, the parameters also play an important role in EAs since parameters determine the search direction and search domain.

There exist precedence constraints among the tasks of the identical mobile device and each task may be executed in a specific determined server. This case study can be regarded as the scheduling of task sequence aiming to minimize the total processing time. As shown in Fig. 2(a), it can be represented as a directed acyclic graph G= (V, A, E), where V is the node set, A represents arcs, and E denotes the disjunctive arcs in the graph. All nodes in V are divided into two categories including the dummy nodes and the task nodes. The set of task nodes consists of two dummy nodes: the start node S and the termination node F. For each mobile device D_j, the start node links the first task node to form the first directed arc, and the last directed arc is formed by the connection of last task node with termination node. The linked task nodes between node S and node F need to be processed with sequence in a predefined order. The pair of numbers around each task node represents the specific determined server, the corresponding processing time unit, respectively. Pairs of disjunctive arcs represent the precedence relationships among tasks of different mobile devices processed on the identical server. We can obtain a subset E' from E, which represents a viable task sequence when the corresponding graph G' = (V, A, E') is acyclic, via obtaining one arc among each pair of disjunctive arcs. As shown in Fig. 3(b), a subset E' contains five tasks, i.e., T_{21}, T_{11}, T_{12}, T_{13}, T_{32}, which are marked with red rectangles. The length of the longest path from the start node to the termination node determines the total processing time of all tasks in mobile devices. The larger the number of nodes is, the more processing time will be cost for the completion of all tasks.



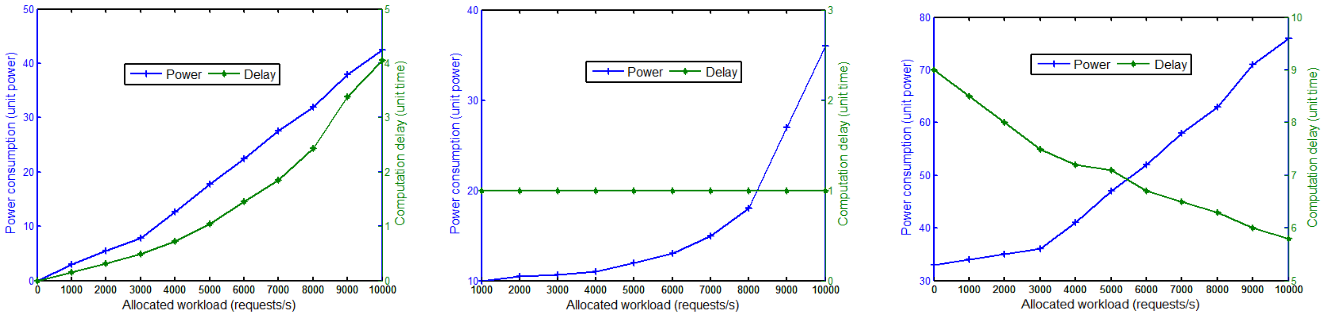
(a)

(b)

Figure 2. An example of task-driven resource assignment under sever-determined condition. a) an illustration disjunctive graph. b) the corresponding Gantt chart. A subset E' contains five tasks, i.e., T_{21}, T_{11}, T_{12}, T_{13}, T_{32}, which are marked with red rectangles.

In Fig. 2(a), D_1 (T_{11}-> T_{12}->T_{13}), D_2 (T_{21}->T_{22}->T_{23}), D_3 (T_{31}-> T_{32}) are three mobile devices processed on three servers, i.e., S_1, S_2, S_3 in MEC system. Each task needs to be processed on the specific determined server with processing time units which is marked as (m,n) around each task node. One feasible task sequence is obtained from Fig. 2(a) and the corresponding Gantt chart is drawn in Fig. 2(b). We can see that the total processing time units are 11.

We evaluate the performance of the task-driven resource assignment for determined sever via the simulation results. In our scenario, experimental settings and parameters are referred from [5], and then we adopt three cloud servers (Internet data centers) and five MEC devices in the MEC system. As shown in Fig. 3(a), when only MEC servers are used for the execution of tasks, the computation delay and power consumption increase simultaneously with the increase of the allocated resources. When only cloud servers are utilized for task execution, Fig. 3(b) illustrates that the computation delay keeps steady with the increase of the allocated workload. Instead, the power consumption increases. From the numerical results in Fig. 3(c), it can be known that the consumption of system power depends on the power consumption of MEC devices, while the communication delay of the WAN has the control over the system delay. This reason is that some tasks are executed in MEC servers, and the system delay decreases with the increasing of the system power consumption. From the analysis of stimulation results, cloud server is more efficient and robust than MEC server. The MEC servers are deployed closed to the mobile users, which is the edge of network. It can be known that if we can sacrifice modest computation resources, the communication delay can be reduced significantly.



(a)

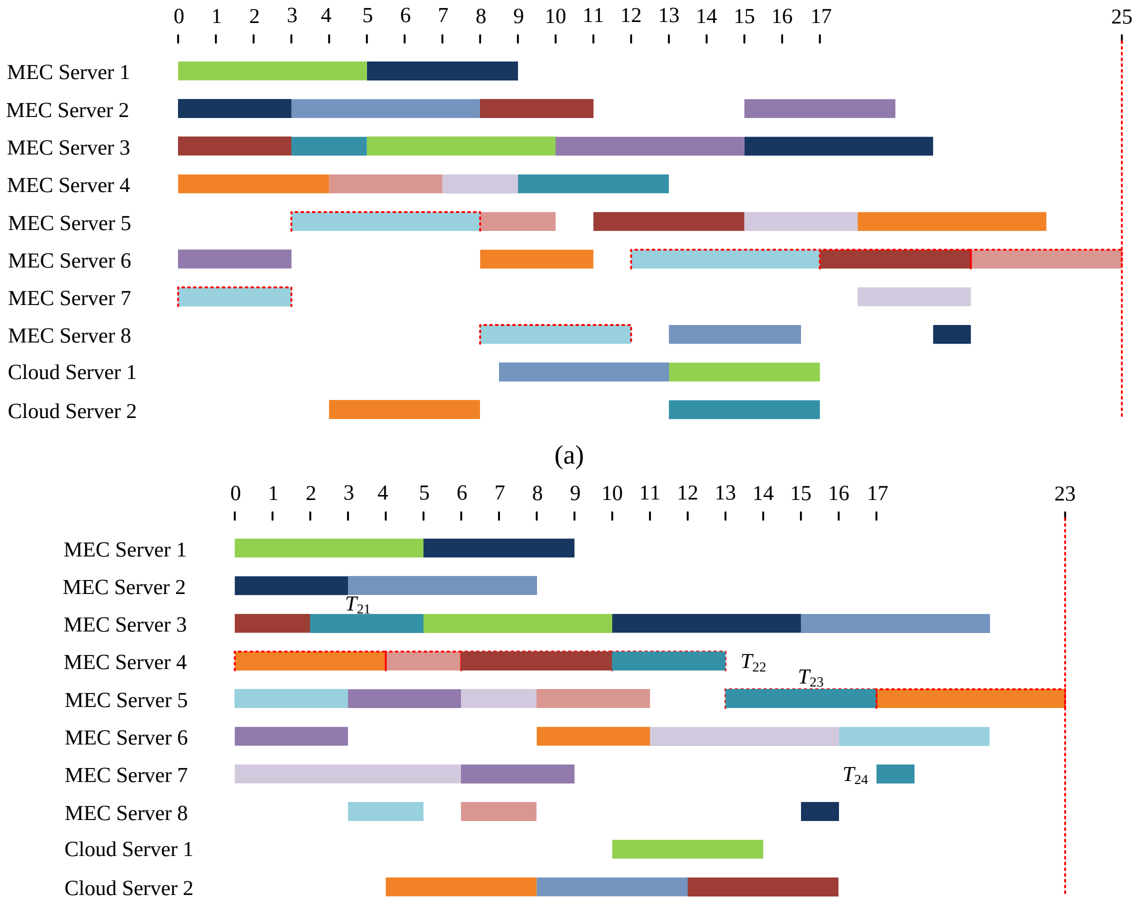
(b)

(c)

Figure 3: Illustration of power consumption-delay tradeoff by task-driven resource assignment for determined sever in a MEC system. a) All tasks are excuted in the MEC servers. b) All tasks are excuted in the cloud servers. c) All tasks are excuted in the MEC system consisting both MEC and cloud servres.

V. Case study II: Server-Flexible Condition

In case I, the total cloud computing delay is consists of two factors: computation delay of cloud computing and communication delay of WAN. Even though the communication delay of WAN is very short, the resource assignment for MEC and cloud servers may be unreasonable under sever-determined condition. In addition, there exists an inevitable problem, which may delay the entire processing. i.e., some servers will not be available in an uncertain time with low probability because of crashing, being repaired, etc. However, the corresponding users always want to obtain the response of users' requirements in short time. *In this case, a server-flexible MEC framework is proposed. i.e., a task can be processed on a set of available servers rather than on one determined server.* The efficiency of resource allocation problem operated in parallel server environment can be well developed, since it is not determined by a predefined order. The communication delay may be decreased significantly, and the robustness of system comparing with case I can be intensified with more flexibility.



(a)

(b)

Figure 4: Gantt charts under two different task-driven resource assignment conditions. a) Server-determined condition. b) Server-flexible condition. Each color stands for one mobile device, and the sub-blocks of each color stand for the execution order of different tasks in one mobile device.

In this case, task-driven resource assignment problem can be viewed as two sub-problems involving task dispatching and resources assignment with the objective to minimize total processing time. e.g., n mobile devices are given and each mobile device contains a series of tasks to be processed in a set of available MEC servers and cloud servers. There exist precedence constraints among tasks of each mobile device, and all mobile devices are independent of each other. Each task must be completed without interruption once it starts.

The combinational optimization consisting of two sub-problems results in an NP-hard problem which means it is difficult to solve. In this case, the hybridation of the classical EAs is preferred because different EAs have various advantages and disadvantages. For example, GA is not easy to fall into local optima, and it is easy to understand without complex math knowledge. However, GA does not perform well in global search. PSO has strong ability of global search while PSO is weak in neighborhood search and cannot be applied for solving problems directly.

In order to verify the performance of task-driven resource assignment for flexible severs, we test our framework with synthetic datasets. The number of mobile devices is set as 10, and the number of tasks for each mobile devices is set as 3 or 4. The number of MEC servers is 8, and the number of cloud servers is 2 in our proposed MEC framework. The Gantt charts of two problems are given in Fig. 4. It can be seen that the computation delay of task-driven resource assignment of server-flexible condition is obviously less than that of server-determined condition. We can draw a conclusion that task-driven resource assignment of server-flexible condition defeats that of server-determined condition in all aspects, such as stability, efficiency, maintainability, etc. However, it cannot be neglected that task-driven resource assignment of server-flexible condition, which consists of two sub-problems involving task dispatching and resource assignment, has higher computational complexity than that of server-determined condition.

VI. Case study III: Server-Uncertain Condition

The idealized computation delay of one task on both MEC and cloud servers is fixed in case I and case II. However, it still needs to be noticed that uncertain factors undoubtedly exist in real MEC framework, e.g., the inserted tasks, the sudden power outages, the unavailability of servers, network failure and uncertain computational delay. The uncertain factors in MEC framework have to be taken into consideration for improving the performance of it. These uncertainties may affect the response times for users, i.e., the final completion time of all tasks which is the most significant criterion in real MEC framework considering users' experiences. In general, the tasks can be offloaded to all servers with the guarantee of internet and electricity. Thus, the uncertain computation delay is the most significant uncertain factors which cannot be ignored.

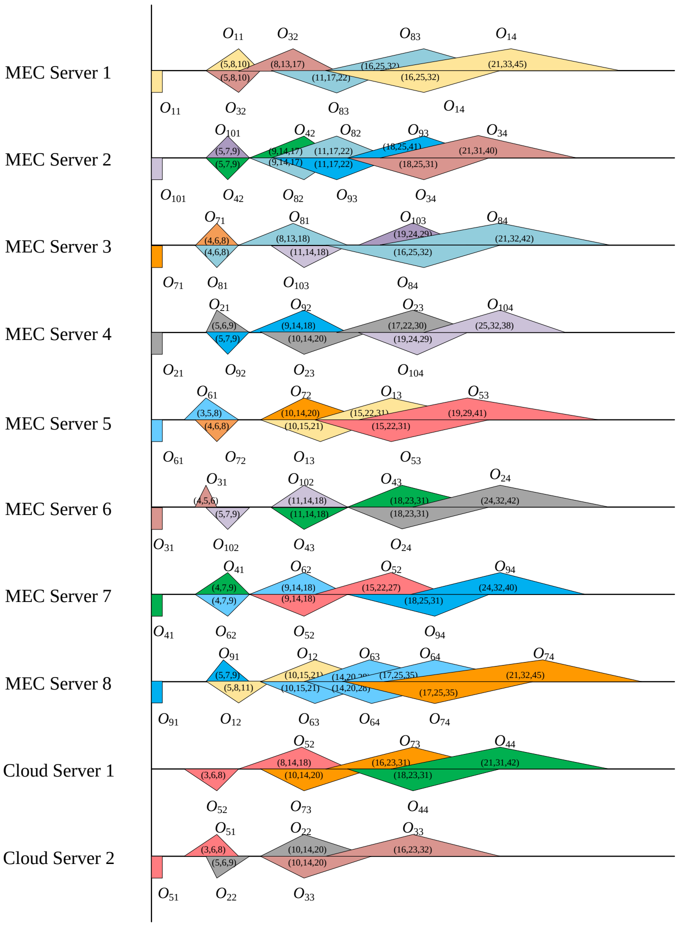


Figure 5: Fuzzy Gantt chart under sever-uncertain condition.

The difference between case III and case II is that the computation delay of each task on corresponding server is uncertain for case III. i.e., the realized output of a fuzzy computation delay of the task only can be known when the whole task is completed. *In this case study, we model the uncertain factor of computation delay based on fuzzy theory which is represented by a triangular fuzzy number*. In fact, as the typical combinatorial optimization problem, the task-driven resource assignment is necessary to combine the mathematical fuzzy number approaches with traditional optimization approaches. Thus novel evaluation criteria, evaluation model and algorithm also need to be designed.

As discussed above, the parameters play an important role for EAs as well, especially for the cases with uncertainty. Therefore, the EAs with parameters self-adaptive mechanism are considered in this case. In traditional EAs, the parameters are manual setting, and they are given in advance before the simulation experiments begin. This setting mechanism is weak in the case with uncertainty because the parameters by manual setting is time consuming. Moreover, the uncertainty cannot be estimated before the cases begin. The ordinary and determined parameters setting cannot help EAs achieve good performance. On the contrary, inappropriate parameter settings may result in worse performance.

In this simulation, we formulate a mixed integer programming model to transform the computation delay into fuzzy variables. The task-driven resource assignment for sever-uncertain condition can be formulated as an extended version of task-driven resource assignment of server-flexible condition, i.e., case II. Each task is carried out with the fuzzy computation delay $\tilde{T}_{mn}$={t_1, t_2, t_3} on the server S_n, where t_1, t_2 and t_3 represent the minimum value, the most-likely value and the maximum value, respectively. The simulation parameters are identical with case II except the computation delay. In the experiment of case III, the computation delay is presented as triangular fuzzy number based on fuzzy theory. The fuzzy Gantt chart under sever-uncertain condition is shown in Fig. 5. It can be known that the total computation delay of all tasks can be predicted even when the computation delay of each task is uncertain. In terms of the task-driven resource assignment of server-flexible condition shown in Fig. 4, if the computation delay of the second assigned task on MEC server 3, i.e., T_{21}, becomes larger, T_{21} needs to be processed behind the completion of the last task on MEC server 3. Then the process of T_{22}, T_{23} and T_{24} have to wait until the completion and release of T_{21}. This leads to the unpredicted increase of total computation delay and the wasted resources produced by the waiting period.

VII. Challenges

The joint deployment on MEC and cloud servers brings various benefits to MEC framework. However, there is still space for researchers to improve or design new paradigms for the task-driven resource assignment in corresponding computational mode and resolve the remaining key challenges for the fast development of MEC.

A. Resource constraints

Compared with cloud servers, the computational and stored resources of MEC servers are limited because of the limited processing ability of their hardware. We expect the computation ability and stored memory in each MEC or cloud server to be infinite and have the scalability to support any application with arbitrary resource requirements. However, this is obviously unrealistic and infeasible. Therefore, the resource constraints for both the MEC and cloud servers should be considered during the process of resource assignment.

B. Large scale

The number of MEC or cloud servers may not satisfy the large requirement of tasks with the restricted computational and storage resources. In real world applications, there are many mobile devices, and various mobile devices can provide various data services, which means that these devices are still growing in quantity. Thus, new framework of evolution computation, which is suitable for distributed computing, should be designed to deal with numerous tasks in mobile devices.

C. Security

Although the combination of MEC and cloud servers for MEC framework has the efficient response for mobile users, the security is still a significant issue that cannot be neglected. The compatible trust information can be exchanged among trust management systems, while MEC framework can perform well even when these trust information belong to different trust domains. This is the potential safety loophole that will harm the MEC trust management systems. Therefore, new encryption mechanism of exchanged information should be performed when designing MEC framework.

VIII. Conclusion

In this article, a task-driven resource assignment framework is proposed for improving MEC system, which integrates both the MEC and cloud servers. Particularly, we study three task-driven cases, sever-determined condition, sever-flexible condition and sever-uncertain condition for resource assignment based on evolutionary computation, while the computation delay of each task on corresponding server is uncertain for the server-uncertain condition. According to different cases, EAs with various mechanisms are discussed, e.g., the hybridation of EAs, the parameters self-adaptive, etc. The performance evaluations on three different conditions are analyzed and compared. We highlight the existing technical challenges and open issues for the safe and stable development of MEC systems, and the encouragement of new paradigms of task-driven resource assignment for MEC systems. We believe that task-driven MEC system will attract more attentions and efforts from researchers before long, and the technologies for integrating MEC servers and cloud servers will be widely developed.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (61801076), Natural Science Foundation of Liaoning Province, China (201602154), Fundamental Research Funds for the Central Universities (DUT17RC(3)029), Dalian Science and Technology Innovation Fund under Grant No. 2018J12GX048.

References

[1] K. Liang *et al*., "An Integrated Architecture for Software Defined and Virtualized Radio Access Networks with Fog Computing," *IEEE Network,* vol. 31, no. 1, 2017, pp. 80-87.

[2] E. Baccarelli *et al*., "Fog of Social IoT: When the Fog Becomes Social," *IEEE Network,* vol. 32, no. 4, 2018, pp. 68-80.

[3], D. Zeng, *et al*., "Joint Optimization of Task Scheduling and Image Placement in Fog Computing Supported Software-Defined Embedded System," *IEEE Trans. on Computers,* vol. 65, no. 12, 2016, pp. 3702-3712.

[4] L. Gu, *et al*., "Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System," *IEEE Trans. on Emerging Topics in Comput.,* vol. 5, no. 1, 2017, pp. 108-119.

[5] R. Deng, *et al*., "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption," *IEEE Internet of Things J.,* vol. 3, no. 6, 2016, pp. 1171-1181.

[6] C. T. Do, *et al*., "A Proximal Algorithm for Joint Resource Allocation and Minimizing Carbon Footprint in Geo-Distributed Fog Computing," *2015 Int. Conf. on Inf. Netw. (ICOIN).*

[7] C. You, *et al*., "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," *IEEE Trans. on Wireless Commun.,* vol. 16, no. 3, 2016, pp. 1397-1411.

[8] J. Liu, *et al*., "Delay-Optimal Computation Task Scheduling for Mobile-Edge Computing Systems," *2016 IEEE Int. Symp. on Inf. Theory (ISIT),* IEEE, 2016, pp. 1451-1455.

[9] T. X. Tran *et al*., "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Commun. Mag.,* vol. 55, no .4, 2017, pp. 54-61.

[10] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic Resource Allocation Exploiting Mobility Prediction in Mobile Edge Computing," *2016 IEEE 27th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Commun. (PIMRC),* 2016, pp. 1-6.

[11] S. Wang, *et al*., "Dynamic Service Migration in Mobile Edge-Clouds," *IFIP Netw. Conf. (IFIP Networking),* 2015, pp. 1-9.

[12] S. Kosta, *et al*., "Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading," *Proc. IEEE INFOCOM*, 2012, pp. 945-953.

[13] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-Efficient Adaptive Resource Management for Real-Time Vehicular Cloud Services," *IEEE Trans. on Cloud Comput.,* 2019, vol. 7, no. 1, 2019, 196-209.

[14] L. Ni, *et al*. "Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets." *IEEE Internet of Things J.,* vol. 4, no. 5, 2017, pp. 1216-1228.

[15] R. Deng, *et al.* "Towards Balanced Energy Charging and Transmission Collision in Wireless Rechargeable Sensor Networks.," *Journal of Communications and Networks*, vol. 19, no. 4, 2017, pp. 341-350.

BIOGRAPHIES

Liangtian Wan [M’15] (wanliangtian@dlut.edu.cn) received the B.S. degree and the Ph.D. degree in the College of Information and Communication Engineering from Harbin Engineering University, Harbin, China, in 2011 and 2015, respectively. From Oct. 2015 to Apr. 2017, he has been a Research Fellow of School of Electrical and Electrical Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor of School of Software, Dalian University of Technology, China. Dr. Wan has been serving as an Associate Editor for IEEE Access and Journal of Information Processing Systems. His current research interests include data science, big data and graph learning.

Lu Sun [S’18] (sunlu517@mail.dlut.edu.cn) received BS and MS degrees at the Dalian University of Technology in 2013 and 2015 respectively and is currently a Ph.D. student at School of Software, Dalian University of Technology. She focuses on computational intelligence, deep learning, probabilistic graphical models, and their applications in combinatorial optimization problems.

Xiangjie Kong [M’13-SM’17] (xjkong@ieee.org) received the BSc and PhD degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor in School of Software, Dalian University of Technology, China. He has served as (Guest) Editor of several international journals, Workshop Chair or PC Member of a number of conferences. Dr. Kong has published over 70 scientific articles in international journals and conferences (with 50+ indexed by ISI SCIE). His research interests include intelligent transportation systems, mobile computing, and cyber-physical systems. He is a Senior Member of IEEE and CCF, and a Member of ACM.

Yuyuan Yuan (yuyuan.yuan@outlook.com) received the B.S. degree from NanChang University, NanChang, China, in 2016. She received the M.S. degree from Dalian University of Technology, China, 2019. Sheis s currently working at CMB Network Technology. Her research interests include recommender systems and social computing.

Ke Sun (kern.sun@outlook.com) received his B.S and M.S and in computer science from Shandong Normal University in 2012 and 2015 respectively. He is currently a PhD candidate student in the School of Software, Dalian University of Technology. His research interests include social science, machine learning, and social computing.

Feng Xia [M'07-SM'12] (f.xia@ieee.org) received the BSc and PhD degrees from Zhejiang University, Hangzhou, China. He was a research fellow with the Queensland University of Technology, Australia. He is currently a full professor in the School of Software, Dalian University of Technology, China. He is the (guest) editor of several international journals. He serves as the general chair, PC chair, workshop chair, or publicity chair of a number of conferences. He has authored two books and more than 200 scientific papers in international journals and conferences. His research interests include data science, big data, knowledge management, network science, and systems engineering. He is a senior member of the IEEE and the ACM.

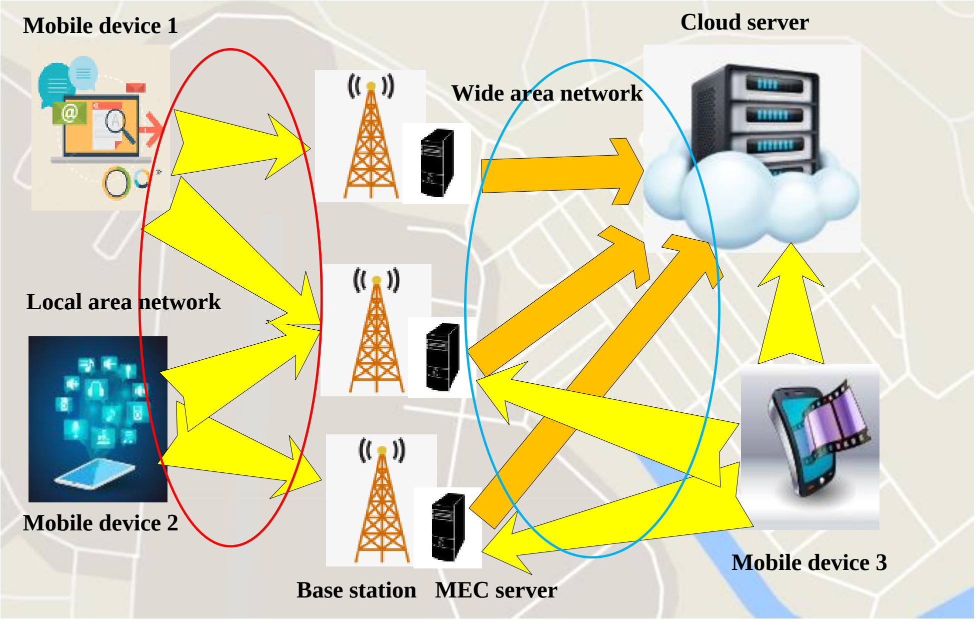
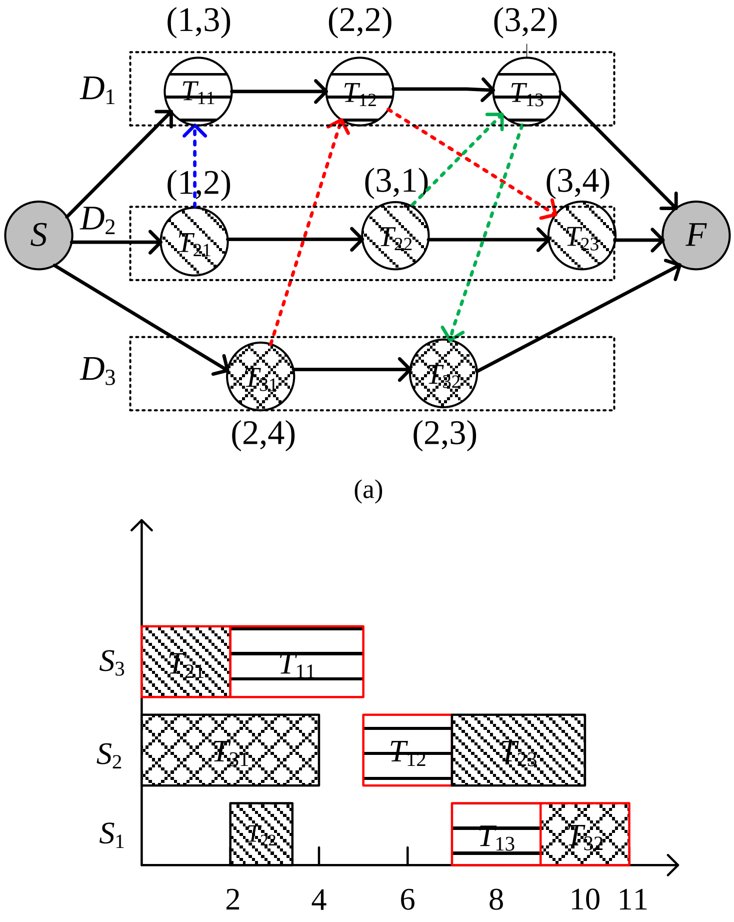


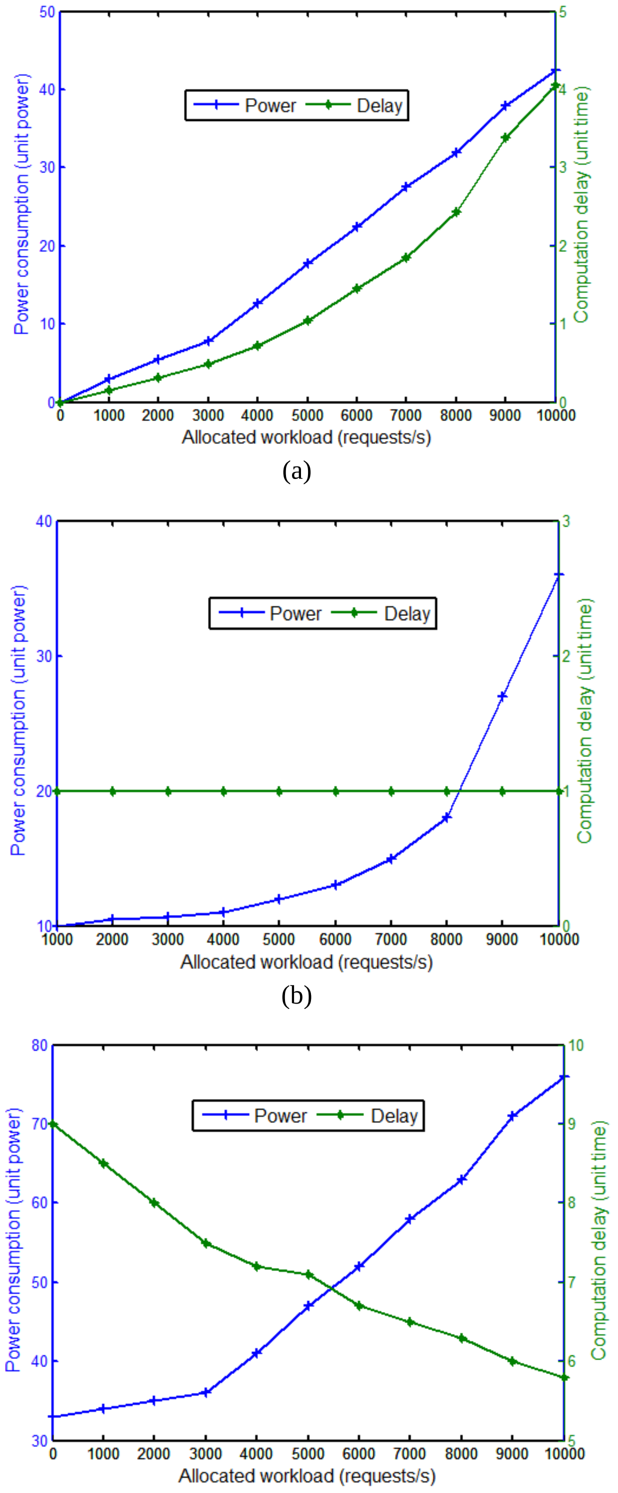
Figure 1: Task-driven MEC framework.



(a)

(b)

Figure 2. An example of task-driven resource assignment under sever-determined condition. a) an illustration disjunctive graph. b) the corresponding Gantt chart. A subset *E*’ contains five tasks, i.e., T_{21}, T_{11}, T_{12}, T_{13}, T_{32}, which are marked with red rectangles.

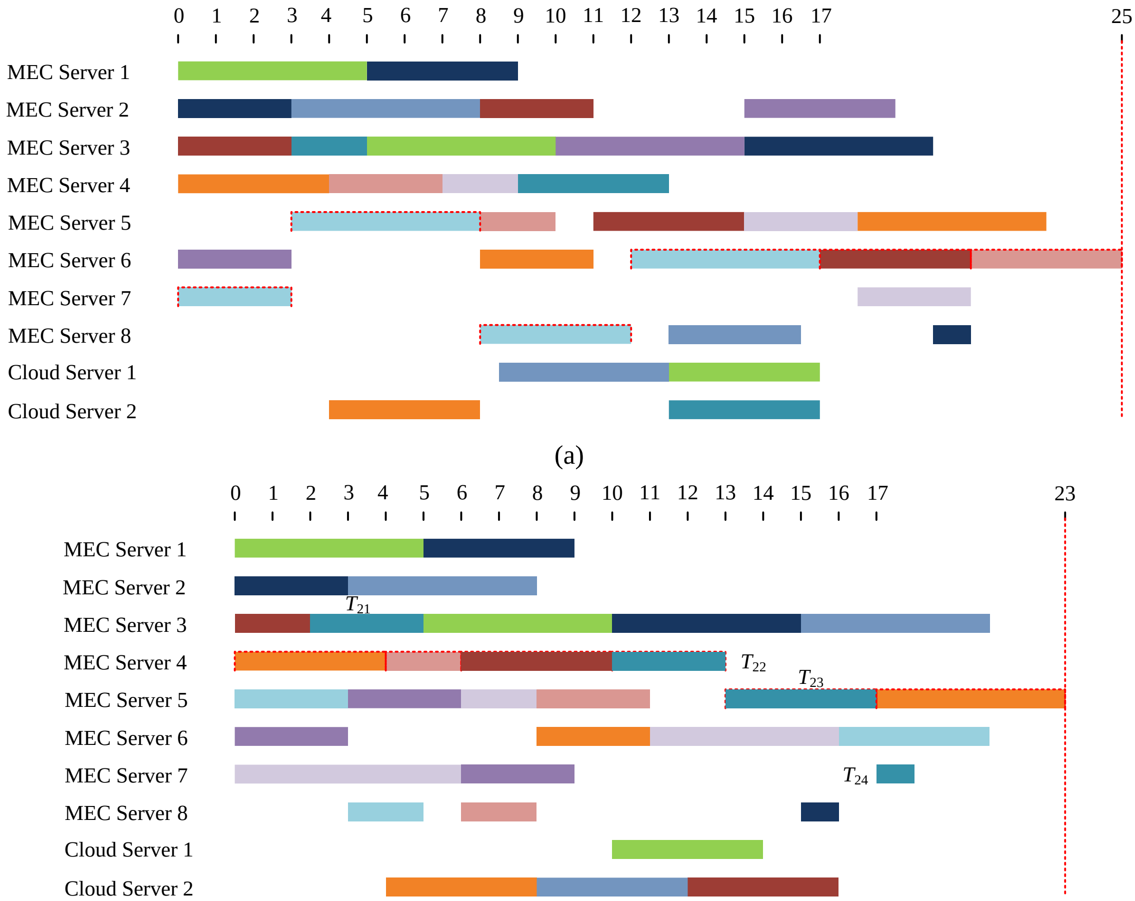


(a)

(b)

(c)

Figure 3: Illustration of power consumption-delay tradeoff by task-driven resource assignment for determined sever in a MEC system. a) All tasks are excuted in the MEC servers. b) All tasks are excuted in the cloud servers. c) All tasks are excuted in the MEC system consisting both MEC and cloud servres.



(a)

(b)

Figure 4: Gantt charts under two different task-driven resource assignment conditions. a) Server-determined condition. b) Server-flexible condition. Each color stands for one mobile device, and the sub-blocks of each color stand for the execution order of different tasks in one mobile device.

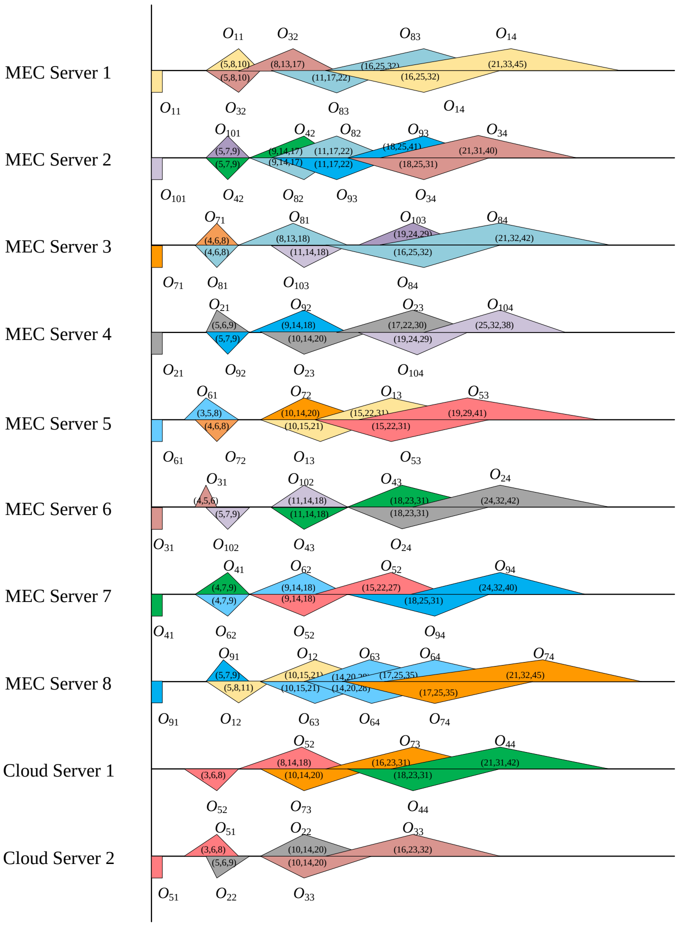


Figure 5: Fuzzy Gantt chart under sever-uncertain condition.