Online Task Allocation and Scheduling in Fog IoT using Virtual Bidding

Nikita Joshi
Dhirubhai Ambani Institute
of Information and Communication Technology,
Gandhinagar
nikita_joshi@daiict.ac.in

Sanjay Srivastava
Dhirubhai Ambani Institute
of Information and Communication Technology,
Gandhinagar
sanjay_srivastava@daiict.ac.in

2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC) | 978-1-6654-0156-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/R10-HTC54060.2022.9929631

***Abstract*—IoT applications increasingly exceed the available computing capacity at the edge. A 3-tier fog architecture may allow for low-latency compute-intensive applications. However, resource allocation in the cloud and fog, along with real-time application requirements, pose a number of challenges. In this paper, we examine the problem of auction-based resource allocation for delay-sensitive online IoT applications in the cloud-fog-edge architecture. We propose a multi-attribute double auction-based task allocation algorithm with delay-based pricing and bidding strategy. A novel virtual-bidding mechanism is designed to allocate resources to tasks that arrive in-between the bidding rounds. The price and winners are determined using McAfee and reserved price-based allocation, which also considers the perishable nature of fog/cloud resources. The proposed algorithm is implemented in NetSim and Python. We find a substantive performance boost in resource utilization when virtual bidding is used for online jobs instead of only using periodic auctions. Simulation results show an increase of 61% in resource utilization, 37% in user utility, and 79% in profit for fog service providers.**

***Index Terms*—IoT, cloud computing, fog computing, resource allocation, double auction, McAfee, QoS, perishable, reserved price, auction**

I. Introduction

The benefits of varied use cases of the Internet of Things(IoT) have increased IoT adoption. IoT applications generate large amounts of time-critical data that must be processed. According to an IDC forecast, an estimated 41.6 billion connected IoT devices or "things" will generate 79.4 zettabytes (ZB) of data by 2025 [1]. However, because of limited resources and energy constraints, IoT devices cannot process these data. In some IoT applications, data is sent to remote cloud servers for processing [2]. Cloud nodes have significant resources to handle IoT data. However, the delay between cloud and IoT devices is not acceptable by many critical IoT applications such as Healthcare [3], Vehicular Adhoc Network(VANET) [4] etc.

In fog computing, IoT devices and processing units are nearer to each other than the cloud. Though fog devices have fewer computing resources, their latency is lower than that of the cloud, making fog closer to IoT devices than the cloud. Several health care IoT applications such as electrocardiogram monitoring (ECG) [5] and hypertension attack detection [6] are using Fog IoT technology.

Cloud and fog service providers(CSPs and FSPs) are now widely available in the market including Amazon, Microsoft [7], and IBM [8]. IoT Service Providers(ISPs) can use them directly instead of deploying resources by themselves. It will increase scalability, resource utilization, and decrease maintenance costs. Resource allocation or task allocation for fog IoT architecture is a known problem in which a fog is found which meets the QoS requirements of particular application. The literature proposes many batch allocation algorithms [9] [10]. However, IoT applications generate online task requests, which means they will generate tasks at any time and task requirements are not known prior to the occurrence. This sporadic nature must be taken into consideration while designing resource allocation algorithms. Additionally, service providers share their resources on a rent basis in a non-monolithic architecture.

Furthermore, resource allocation must take into account monetary costs. Hence, we need a resource allocation algorithm that considers IoT applications' non-monolithic architecture and online nature. Determining service providers' prices is also a big challenge for this architecture. Auction mechanism is the best method for determining price [11]. Based on a player's bidding strategy, there are three types of auction mechanisms. The first is forward auctions, in which users bid for resources, and the service provider selects the highest paying users so it can maximize its profit [12]. The second method is a reverse auction, where service providers bid and users select the lowest asking service provider [13]. Lastly, a double auction mechanism is used, where the user submits a bid and the service provider submits their requests to the auctioneer, and the auctioneer finds the winner and the final price for both [4]. In Fog IoT architecture, the double auction mechanism is most appropriate as users have a budget and fog nodes have a price requirement equal to their maintenance costs.

In any auction mechanism, the payment mechanism must guarantee truthfulness, budget balance, and individual rationality [11]. The following payment mechanisms have been utilized for double auctions: VCG based, McAfee, and K-double auction [14]. While VCG ensures truthfulness, it does not ensure budget balance, which means that auctioneers may occasionally have negative utility. McAfee guarantees all these three properties [15]. We used the extended McAfee algorithm to determine prices in our work.

This paper is organized as follows: Related work is discussed in Section II. The system model is explained in section III. Followed by the formulation of the problem statement in section IV. The Task allocation algorithm is explained in section V. Finally, Simulation results are discussed in section VI with conclusions.

II. Related work

In recent studies, the issue of resource allocation in Fog IoT architecture has been addressed. This comparative analysis of the literature studies have been summarized using three parameters in Table I. The first parameter to consider is the allocation cycle. The allocation can be done in batch mode, where multiple requests are collected before executing the allocation algorithm. Another approach is an online mode, in which tasks are allocated as they arrive without waiting for other tasks to arrive. A second parameter is whether or not QoS requirements of IoT applications are taken into account when allocating resources. A third parameter considered for this classification is whether architecture is monolithic or non-monolithic. When allocating resources in non-monolithic architectures, the payment mechanism needs to be considered.

TABLE I: Classification of literature
B:Batch, O:Online, M:Monolithic, N:Non monolithic

Paper	Allocation cycle	QoS	Architecture
[9] , [16] , [17]	B	✓	M
[10], [12]	B	×	N
[13] [4]	B	✓	N
[18]	O	✓	M
[19], [20]	O	✓	N
[21]	B/O	✓	N

Studies including [9] [16] and [17] have studied periodic resource allocation in fog IoT architectures, targeting IoT requirements such as deadline, priority, energy minimization, and delay minimization. Monolithic architecture, where a single entity controlled all resources at the fog and cloud levels, was considered. As a result, the price is not considered while allocating tasks. [10] and [12] have emphasized only money-based allocation in fog computing without considering QoS requirements.

Agrawal et al. [13] propose reverse auction-based resource allocation for the IoT fog, which also meets the requirements for QoS in IoT applications. Only service providers are permitted to bid in this reverse auction mechanism. Therefore, the budget of users is not considered while allocation. Peng at al. [4] has proposed a double auction-based resource allocation method that creates a bipartite graph of users and fog nodes. Edges in the graph are weighted based on QoS requirements. Allocation is then performed using maximum matching of a bipartite graph. The extended McAffee method is used to find the final price for buyers and sellers. Matching is used to find allocation, so a fog can serve only one user, even if some of its resources can satisfy the needs of other users. Additionally, all these studies use batch allocation only.

Research work including [18], [19], [20] have proposed online resource allocation in fog IoT architecture. Haisheng et al. [18] have minimized the weighted response time of job execution by using preemption-based scheduling policy and priority-based job dispatching policy. The study, however, considered monolithic architecture, so pricing mechanisms were not examined.

Jie et al. [19] has divided time into rounds. Users submit requests within the designated rounds. Repeated Stackelberg game is used to find the optimal solution. However, they have assumed that only one user will send a request in a round, and each user's turn is fixed. As a result, it is not an actual online allocation where resources can be requested randomly by any user. This study does not address payment mechanisms for FSPs and ISPs. Nguyen et al. [20] have assumed that a set of buyers are fixed and have registered with brokers to obtain resources. Sellers may appear and disappear at any time. A broker allocates a user to the seller so that the user's utility is maximized, but the fog service provider's utility is not maximized. Neither of these studies has an auction mechanism been employed to determine the price of resources.

In our previous work [21], the Stackelberg game was used to allocate periodic and sporadic tasks and determine the price for CSP. However, the limitation of the Stackelberg game is that it is a leader-follower game. CSP acts as a leader and decides price and quantity. The user's budget is not taken into account. Additionally, FSPs are not able to share the cost of resources. In order to maximize the profit of both parties, FSPs and ISPs, the auction mechanism is best for determining price since it takes advantage of the market conditions to determine the price. However, a traditional auction mechanism requires that all buyers bid before allocation is started. At a time, online task allocation brokers receive bids from only one user, which makes it hard to decide whether to accept a bid or not.

Zang et al. [22] , Gharaibeh et al. [23] have proposed online auction based task allocation for cloud resources. In which, only one service provider is considered, so we need to consider competition among ISPs only, but in fog computing architecture, we have multiple FSPs; therefore, we have competition among ISPs as well as competition among FSPs. The resource price should be determined by considering both the competition and demand-supply situation of the market. The fog/cloud resources are perishable, which means they cannot be accumulated and reused in the future if they are not allocated at a time. In our previous work [24] we have proposed double auction based task allocation mechanism which maximizes utility of system and task success ratio as well as resource utilization. We have extended that work in this paper with considering perishability of resources and designed bidding strategy as well as pricing mechanism. To the best of our knowledge, no previous research work has evaluated task allocation in batch mode and online mode using non-monolithic architectures and auction mechanisms that consider competition between FSPs and ISPs, IoT QoS requirements and resource perishability.

III. System Model

Non-monolithic three-tier architecture is used in this research. There are N ISPs, M FSPs and one CSP available. Each FSP registers with the CSP by sending $I_j =< a_j, \mu, E_j >$, where a_j is available Virtual machines(VMs), μ is the service rate of one VM, and E_j is maintenance cost per one unit of VM. When ISP needs to execute a task, it sends task requirements to CSP. CSP acts as broker here. CSP finds appropriate FSP and informs ISP. Task requirement contains λ_i, p_i, D_i^s and D_i^h where λ_i is required service rate, p_i is priority, D_i^s is soft deadline and D_i^h is hard deadline of task from i^{th} user. This model assumes that the task arrival rate follows the Poisson distribution.

Quantity of VMs to be purchased is determined by Eq. (1)

$$q_i = \frac{\lambda_i}{\mu} \tag{1}$$

Delay between i^{th} ISP to j^{th} FSP/CSP includes network delay η_{ij} and service delay s_j which is represented in Eq. (2)

$$d_{ij} = \eta_{ij} + s_j \tag{2}$$

s_j includes queuing delay and execution delay e_i. To calculate queuing delay, service is modeled using M/M/c queuing model where c equal to number of VMs at FSP. Queuing delay is calculated as waiting time in queue or M/M/c queue [25].

$$e_i = \frac{1}{\mu} \tag{3}$$

Based on the cost of each node(FSP/CSP/ISP) and payment received, we have defined the utility function of each node. ISPs provide service to IoT customers. Therefore, ISPs get revenue from IoT customers to execute each task. This revenue is called valuation of the task for ISP. Additionally, ISP needs to pay CSP. Therefore, its utility function is defined as follows:

$$U_i^h = V_i - C_i \tag{4}$$

where V_i is valuation of task for user which is calculated by

$$V_i = (p_i\alpha_{i1} - \alpha_{i2}\frac{c_t - D_i^s}{c_t})q_i \tag{5}$$

where p_i is priority of task, q_i is amount of VMs required by task,α_{i1} is revenue generated from IoT customer per unit VM, D_i^s is soft deadline of task, c_t current time of the system. for IoT application valuation of task decreases linearly with time greater than soft deadline.

$$\alpha_{i2} = \begin{cases} 0, & \text{if } c_t < D_i^s \\ 1, & \text{else} \end{cases}$$

CSP acts as auctioneer here. It also has resources that can be used to execute tasks if FSP resources are not available. The utility of CSP can be calculated as follows:

$$U^c = \sum_{i=1}^{N}(C_i - \tau_{i,0}\theta_i - L_i) - \sum_{j=1}^{M} F_j \tag{6}$$

Here $\tau_{i,0}$ is 1 if task is allocated to cloud itself. θ_i is energy cost to execute task of i^{th} user at cloud. F_j is total payment of j^{th} FSP. L_i is penalty charged by user if task is not completed before deadline which is same as used in [21].

FSP receives payment from CSP and it has maintenance cost to maintain these resources either their used or not used. Utility function of fog node is defined as

$$U_j^f = F_j - a_jE_j \tag{7}$$

where E_j is maintenance cost per one VM and F_j is total payment received from CSP.

$$F_j = \sum_{i=1}^{N} q_i\tau_{ij}r_j \tag{8}$$

where r_j is price of j^{th} FSP for one unit of VM.

IV. Problem statement

The main objective of this research work is to design an algorithm which performs deadline and priority based allocation with following objectives:

$$\text{Maximize} \qquad \sum_{i=1}^{N}\sum_{j=1}^{M} U_i^h, U_j^f, U_c$$

subject to

$$\begin{aligned} t_i \leq D_i^h,\ i = 1, \ldots, N. \\ U_i^h \geq 0,\ i = 1, \ldots, N. \\ U_j^f \geq 0,\ j = 1, \ldots, M. \\ U_c \geq 0 \end{aligned}$$

This problem is a case of the joint optimization problem. It is not easy to maximize the utility of all three entities simultaneously [26]. Additionally, we have considered the online nature of IoT applications. The task arrival rate follows Poisson distribution, and task requirements are unknown, making this problem more challenging. Therefore we have proposed a heuristic-based algorithm that uses a multi attribute-based double auction mechanism to allocate resources.

V. Proposed work

We propose a multi-attribute double auction based mechanism. We will allocate tasks in two modes namely batch mode and online mode. In both modes, we have to first, find an FSP best matched with the ISP's QoS characteristics and then find the final price that should be paid by the ISP and the final payment that should be made to FSPs. In our system, time is divided into rounds. At the beginning of each round, batch allocations are performed. Suppose CSP receives a task request. The next round of batch allocation starts after b_t seconds and current time is c_t.Tasks can be assigned in batch mode if

$$(b_t - c_t) + e_i \leq D_i^s \tag{9}$$

else online task allocation is performed.

Each ISP sends task requirement vector $Q_i =< \lambda_i, p_i, D_i^s, D_i^h >$ and bid b_i^u for that task to CSP.

$$b_i^u = \begin{cases} \frac{V_i}{q_i} r^s, & \text{if } r^s <= 1 \\ \frac{V_i}{q_i}, & \text{else} \end{cases} \tag{10}$$

where r^s is resource scarcity of previous round which is ratio of demand and supply

$$r^s = \frac{\sum_{i=1}^{N} q_i}{\sum_{j=1}^{M} a_j} \tag{11}$$

The bid of FSPs are calculated by

$$b_j^f = \begin{cases} E_j r^s, & \text{if } r^s >= 1 \\ E_j, & \text{else} \end{cases} \tag{12}$$

A. Batch mode allocation

1) Mapping of ISPs and FSPs: For batch mode allocation, we have modified the algorithm used in [4]. A bipartite graph is created, where one partition consists of all ISPs and the other partition consists of FSPs, as explained in Algorithm 1. There is an edge between an i^{th} ISP and j^{th} FSP if $a_j > q_i$. We calculate each edge's weight based on the following equation:

$$w_{ij} = \frac{p_i}{p_{max}} + \frac{1}{D_i^s} + \frac{1}{d_{ij}} \tag{13}$$

. We select the highest weighted edge from the graph, which gives eligible pair of ISP and FSP. All other edges on that ISP vertex are deleted if we select an edge. Algorithm 1 is fed an input of ISP requirements vector Q, Fog information vector I, and delay between FSPs and ISPs d. The algorithm returns an allocation matrix W.

Algorithm 1: Mapping of ISPs and FSPs: Batch mode

Input: Q, I, d
Output: W
W=-2
Create G(V,E) where $(i,j)\epsilon E$ if $a_j > q_i$
W$_{ij}$ = is calculated by Eq. (13)
$U = \{v\epsilon V | (v,x)\epsilon E, \exists x \epsilon V\}$
while $U! = \phi$ **do**
  e$_{ij}$ = max weighted edge of the graph
  for *all g=(i,x)$\epsilon E | x\epsilon V$* **do**
    E=E-g
    $W_{ij} = 1$

2) Winner determination and pricing: Upon receiving eligible ISP-FSP pairs, we run a final price determination Algorithm 2 which finds the final winners and the price that ISPs are to pay, as well as the amount paid to the FSPs. We first find winners and payment of ISP and FSP by using the algorithm mentioned in [27] named Winner Determination and Pricing (WDP). The remaining eligible fog user pairs are then assigned if the bid of ISP exceeds the reserve price of FSP. This ISP-FSP pair is then declared the winner, and the price to be paid is equal to the reserve price of FSP. The Reserved price of FSP is equal to the maintenance cost of FSP (E_j).

Algorithm 2: WDPB: Winner determination and price Batch mode

Input: Q, I, τ
Output: τ, C_i, r_j
$\tau = -2$
$\tau, C_i, r_j = WDP(Q, I, \tau)$
for *each user i* **do**
  if $\tau_i == -2$ **then**
    for *each fog j* **do**
      if $W_{ij}! = -2 and E_j < b_i^u$ **then**
        $\tau_i = j$
        $C_i = E_j$
        $r_j = r_j + E_j$

B. Online allocation

We perofrmed virtual auctions for online allocation, in which we collected bids from ISPs for different scenarios so that when the actual task request arrives, we can compare those bids.

1) Virtual auction: It is executed at the beginning of the system and is repeated when a new ISP joins or any ISP leaves the system. ISP i registers with a CSP. The user submits b_{p,d_q^i,r^s} which is the bid of the ISP, p is the priority level, and d_q is the deadline quotient, which is calculated by the formula

$$d_q = \frac{D_i^s - a_t}{D_i^s} \tag{14}$$

Each ISP submits a bid for a different combination of parameters. This bid is decided using Eq.(10). CSP already has E_j for each FSP. Using that CSP calculates b_j^f using Eq. (12). After that, CSP runs Algorithm 2 for all scenarios; it determines the final winning bid and payment for FSP for each scenario.

2) Allocation: The actual online task request is executed in this phase, in which the most eligible fog node with minimum d_{ij}, $a_j > q_i$ and $d_{ij} < D_i^s$ is selected. Next, we need to find the best match scenario from a virtual auction table. The problem was modeled as a classification problem, and we used K-Nearest Neighbor(KNN) [28] to find the best match scenario. Once the best match is determined from the table and its price b^c, task allocation occurs if price b_i^u exceeds b^c.

VI. Simulation and Results

This algorithm has been implemented in python and NetSim. In NetSim, we have created three wireless networks, each of which has an FSP. Each of these wireless networks is connected to a wired node. In every network, FSPs have a minimum delay for devices belonging to that network which acts as ISPs. FSPs can also serve devices belonging to other networks. In NetSim, the following parameters are used: In

TABLE II: NetSim Parameters for Algorithm Implementation

Parameter	Value
Number of sensors	10
Number of routers(fog nodes)	3
Number of wired node(cloud)	1
Packet size	1000 KB
Inter packet arrival time	1 s
Simulation time	200 s

NetSim, all other parameters are set to default. We have derived the round trip time(RTT) between the ISP and each FSP and CSP using NetSim. We have sent 200 packets for each application, 10 of which are requests for allocation packets, and 10 are execution packets based on which utility values are calculated. There are 100 batches of batch allocation, with random online task requests from ISPs ranging from 0 to N in each round. The parameters set in the Python code are listed in Table III.

TABLE III: Simulation parameters

Parameter	Value
A_j	300
p_i	1,2
D_i^s	normal(75000,7500)ms
α_{i1}	random(300,1000)
E_j	normal(300,1000)
μ	normal(0.1,0.001)
k	3

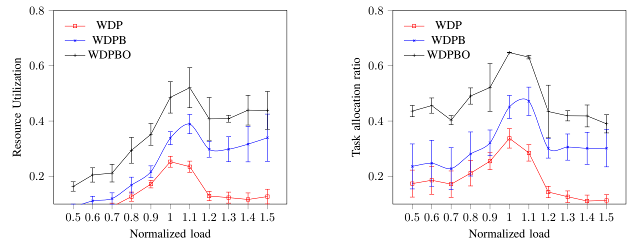
The following algorithms have been compared:

1) Winner determination and pricing(WDP): Batch allocation method used in [29], which uses an extended MCafee algorithm.
2) Winner determination and pricing batch mode (WDPB): This is a batch allocation of our algorithm, consisting of Algorithm 1 and Algorithm 2. Online tasks are allocated in the next batch allocation round
3) Winner determination and pricing batch and online mode(WDPBO): Batch allocation is performed periodically, and online allocation is used for urgent tasks.

We have used normalized load(γ) calculated by Eq. (15) to generate simulations scenario

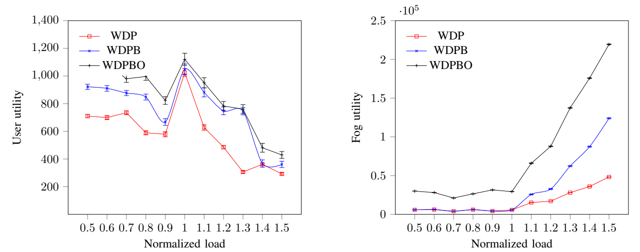
$$\gamma = \frac{\sum_{i=1}^{N} q_i}{\sum_{j=1}^{M} a_j} \tag{15}$$

γ is ranged from [0.5,1.5] which covers low demand scenario to high demand scenario.



(a) Resource utilization Vs. Normalized load
(b) Task allocation ratio VS. Normalized load

Fig. 1: Resource utilization and Task allocation ratio



(a) User utility Vs.Normalized load
(b) Fog utility VS. Normalized load

Fig. 2: User utility and Fog utility

These three algorithms are compared based on the following parameters:

1) Task allocation ratio: That is the ratio of tasks allocated and total task requests received

$$T = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \tau_{ij}}{\sum_{i=1}^{M} f_i} \tag{16}$$

Here, f_i is binary variable. It has a value one if $i^t h$ ISP has generated request; else, it is 0.

2) Resource utilization (Ru_i): It is ratio of total resources allocated and total resources available

$$Ru_i = \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{\tau_{ij} q_i}{A_j} \tag{17}$$

3) Fog utility: Average utility of each fog calculated from Eq. (7)
4) User utility: Average utility of each user calculated from Eq. (4)

Fig. 2 indicates that resource utilization and task allocation ratio increase as normalized load increases. As demand exceeds total capacity, the price of FSP increases, making it difficult for resources to be allocated, thereby reducing resource utilization and task allocation ratio. WDPB allocates remaining resources at maintenance prices, WDPB allocates more tasks than WDP. When the task is allocated in the next batch, sometimes these tasks expire before the new batch starts, so WDPBO has more resource utilization and task allocation ratio than WDPB.

According to Fig. 3, user utility decreases and fog utility increases as demand increases since users need to pay more as demand increases.WDPBO has higher utility than WDP because online tasks are assigned as they arrive, so they are completed before the deadline, resulting in less penalty. WDPB has a higher user utility than WDP because it allocates remaining resources at reserve prices. Therefore, users receive resources at a lower price than fog bid. WDPBO's fog utility is more than WDPB's since it allocates more resources by handling online tasks on time. WDPB's fog utility is more than WDP since it allocates remaining resources at a reserved price. A detailed comparison of WDPBO with WDPB and WDP is shown in Table IV.

TABLE IV: Comparison of WDPBO with WDP and WDPB

Parameter	WDP	WDPB
Task allocation ratio	+59.69%	+34.30%
Resource utilization	+61.23%	+31.63%
Fog utility	+79.28%	+57.28%
User utility	+37.83%	+18.79%

VII. Conclusion

IoT applications benefit from low-latency fog computing as an alternative to cloud processing. In this paper, we have examined the problem of auction-based resource allocation in fog IoT architecture for delay-sensitive IoT applications. We have defined bidding strategies that consider the delay sensitivity of IoT applications and the current demand-supply ratio. A novel virtual-bidding mechanism determines the price requirement for online tasks. We have done batch mode as well as online mode allocation. McAfee and the reserved price determine the final winner and price. We compared our algorithm with the algorithm used in [4](WDP) and with only batch mode allocation(WDPB). Results indicate that resource utilization is improved by 59.69% compared to WDP and 34.30% compared to WDPB. User utility is also improved by 37.83% compared to WDP and 18.79% compared to WDPB. Fog utility is improved by 79.28% compared to WDP and 57.28% compared to WDPB.

References

[1] "IoT data," https://www.businesswire.com/news/home/20190618005012/en/The-Growth-in-Connected-IoT-Devices-is-Expected-to-Generate-79.4ZB-of-Data-in-2025-According-to-a-New-IDC-Forecast, 2022, [Online; accessed 18-03-2022].

[2] M.-k. Suh, C.-A. Chen, J. Woodbridge, M. K. Tu, J. I. Kim, A. Nahapetian, L. S. Evangelista, and M. Sarrafzadeh, "A remote patient monitoring system for congestive heart failure," *Journal of medical systems*, vol. 35, no. 5, pp. 1165–1179, 2011.

[3] Y. Cao, S. Chen, P. Hou, and D. Brown, "Fast: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*. IEEE, 2015, pp. 2–11.

[4] X. Peng, K. Ota, and M. Dong, "Multiattribute-based double auction toward resource allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3094–3103, 2020.

[5] O. Akrivopoulos, I. Chatzigiannakis, C. Tselios, and A. Antoniou, "On the deployment of healthcare applications over fog computing infrastructure," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2017, pp. 288–293.

[6] S. K. Sood and I. Mahajan, "Iot-fog based healthcare framework to identify and control hypertension attack," *IEEE Internet of Things Journal*, 2018.

[7] "Microsoft Azure," https://azure.microsoft.com/en-in/, 2019, [Online; accessed 10-4-2019].

[8] "IBM cloud," https://www.ibm.com/internet-of-things, 2019, [Online; accessed 10-4-2019].

[9] Y. Sahni, J. Cao, and L. Yang, "Data-aware task allocation for achieving low latency in collaborative edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3512–3524, 2018.

[10] X.-Q. Pham, T.-D. Nguyen, V. Nguyen, and E.-N. Huh, "Joint node selection and resource allocation for task offloading in scalable vehicle-assisted multi-access edge computing," *Symmetry*, vol. 11, no. 1, p. 58, 2019.

[11] G. Baranwal, D. Kumar, Z. Raza, and D. P. Vidyarthi, *Auction based resource provisioning in cloud computing*. Springer, 2018.

[12] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *2017 IEEE international conference on edge computing (EDGE)*. IEEE, 2017, pp. 47–54.

[13] A. Aggrawala, N. Kumarb, D. P. Vidyarthic, and R. Buyyad, "Multi-attribute combinatorial reverse auction model for resource procurement in fog integrated cloud architecture."

[14] K. Miyashita, "Online double auction mechanism for perishable goods," *Electronic Commerce Research and Applications*, vol. 13, no. 5, pp. 355–367, 2014.

[15] A.-L. Jin, W. Song, P. Wang, D. Niyato, and P. Ju, "Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing," *IEEE Transactions on Services Computing*, vol. 9, no. 6, pp. 895–909, 2015.

[16] H. Shah-Mansouri and V. W. Wong, "Hierarchical fog-cloud computing for iot systems: A computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[17] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M.-T. Zhou, "Meets: Maximal energy efficient task scheduling in homogeneous fog networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4076–4087, 2018.

[18] H. Tan, Z. Han, X.-Y. Li, and F. C. Lau, "Online job dispatching and scheduling in edge-clouds," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[19] Y. Jie, X. Tang, K.-K. R. Choo, S. Su, M. Li, and C. Guo, "Online task scheduling for edge computing based on repeated stackelberg game," *Journal of Parallel and Distributed Computing*, vol. 122, pp. 159–172, 2018.

[20] D. T. Nguyen, L. B. Le, and V. Bhargava, "Edge computing resource procurement: An online optimization approach," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE, 2018, pp. 807–812.

[21] N. Joshi and S. Srivastava, "Task allocation in three tier fog iot architecture for patient monitoring system using stackelberg game and matching algorithm," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2019, pp. 1–6.

[22] J. Zhang, N. Xie, X. Zhang, and W. Li, "An online auction mechanism for cloud computing resource allocation and pricing based on user evaluation and cost," *Future Generation Computer Systems*, vol. 89, pp. 286–299, 2018.

[23] A. Gharaibeh, A. Khreishah, M. Mohammadi, A. Al-Fuqaha, I. Khalil, and A. Rayes, "Online auction of cloud resources in support of the internet of things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1583–1596, 2017.

[24] N. Joshi and S. Srivastava, "Auction-based deadline and priority enabled resource allocation in fog-iot architecture," in *2020 Futuristic Trends in Network and Communication Technologies(FTNCT)*. Springer, 2020.

[25] O. Kella and U. Yechiali, "Waiting times in the non-preemptive priority m/m/c queue," *Stochastic Models*, vol. 1, no. 2, pp. 257–262, 1985.

[26] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.

[27] D. Yang, X. Fang, and G. Xue, "Truthful auction for cooperative communications," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2011, pp. 1–10.

[28] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, pp. 13–23.

[29] X. Peng, K. Ota, and M. Dong, "Multiattribute-based double auction toward resource allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3094–3103, 2020.