IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

DEADLINE-SENSITIVE TASKS USING AUCTION-BASED VM ALLOCATION IN DISTRIBUTED EDGE CLOUD.

Shivaa Sneha Sri, V.Uma Rani
Student, Associate Professor
School Of Information Technology,JNTUH

Abstract: Edge cloud networking becomes the modern concept where only centralised cloud data centre storage and distribution facilities were transferred towards Edge Cloud Nodes (ECNs) mostly on edge of the network. Comparison to standard cloud data centres, ECNs were physically similar towards smartphone customers such that contact distance was massively diminished. Throughout this article, they research the topic including its distribution for Virtual Machine (VM) assets throughout geo-distributed ECNs for smartphone users utilizing marketplace methodology. Next, they consider smartphone users as well as ECNs like sellers and buyers for VM property auctions, accordingly. After which they design a VM resource allocation algorithm like an n-to-one balanced bipartite graph representing the 0-1 knapsack restriction issue. Although the issue was NP-hard, they are constructing the biased optimal solution for decide the champions of auctions, mostly on basis for that we are proposing the veritable auction-based VM resource allocation (AVA) method for overcome the issue. In addition , we show that perhaps the AVA method not just to provides the essentially ideal outcome to winning choice, yet has the characteristics for truthiness, human rationality as well as computationally efficient. Eventually, they carry out detailed experiments of actual residues for validate the substantial efficiency of proposed AVA system.

***Index Terms* - Auction mechanism, edge cloud, mobile cloud computing, virtual machine allocation.**

I. INTRODUCTION

Recent years have witnessed the spread of mobile computing that uses network people to navigate multiple approaches from everyones connected phones with distant location cloud data centres like Amazon EC2[4], Microsoft Azure[18], etc. Along with leveraging assets (like computing , storage, and so on.) throughout clouds, such connected phones could smash via limited resources for total demanding tasks and greatly improve everyones capabilities. Even so, while phone devices were typically further apart of distant cloud data centres, relocation of apps would probably result to a lengthy pause throughout connectivity and a network capacity, significantly limiting the usability of mobile users. The modern cloud computing model , called the edge cloud[8, 13, 21, 24, 25], often known as drones rather than cloudlets[14, 15, 17, 32], was suggested for alleviate these harmful effects. The variety for small-scale computational and storage servers were located along network edges as shape several EdgeCloud Nodes (ECNs) throughout the edge cloud paradigm. Iphone users should easily connect local ECNs such that bandwidth as well as network loads has been greatly minimized.

The standard Mobile Edge Cloud (MEC) architecture was associated with a wide number of wireless apps, few ECNs, as well as a Centralized Cloud (CC). The ECNs introduced by such cloud providers were spread only at edge of networks. Cell devices may link to ECNs across a wireless wide area network[2, 11, 17] (often defined mostly as access networks network[2, 3, 31], the cellular regional area network[14, 24], etc.), although such ECNs link to just the central CC through a core network[2, 8] (often described mostly as backbone / backhaul network[11, 13, 25], the broad region network[8, 17, 31], etc.), which can be seen in Fig. 1. The ECNs were locally dispersed as well as small-scale relative mostly to distant CC. Any mobile phone users then can provide much for such a cloud service to such an ECN. Instead, unless the ECN fails to produce cloud services to such mobile users, this could submit some of its applications to the CC through the core network, that will eventually lead to this very significant wait throughout communication.

Virtual Machine (VM) resource consumption was among the most critical problems for edge clouds [12, 13, 15, 22, 30]. Notice how some smartphone devices include many cloud computing programmes (that is activities) which have to be tackled. Even so, such consumers do not have ample data as well as computational capacity. We therefore need to rent VM services again from ECNs. In order to secure the consistency for cloud computing implementations, the propagation delay for every assignment should be no greater than that of a time limit (called time limits). Meantime, an ECNs could accomplish their assigned work unless the overall VM resources required through mobile devices were not greater than that of the VM resources of the ECNs (called competitive constrictions). Even so, unless the storage limitation including its ECN will never be met, the ECN could even submit portion of its project given to just the CC within the time limit. Throughout this paper , we focus mostly on distribution of VM resources besides time-limited cloud services assignments throughout the centralised cloud system, hoping to prevent work overload as well as wasting resources from any of the ECNs. In addition, severe competition besides VM resources among customers, we are studying the issue of its allocation with VM assets throughout ECNs with phone devices via the use of auction concept. Throughout fact, the layout of an auction-based VM allocation of resources method throughout MEC was really complicated. Those who are summarising three key issues.

Initially, unlike the conventional two-layer centralized cloud, the edge computing cloud seems to be currently the three-layer framework, which is shown in Fig. 1. Whenever the demanded VM assets surpass the constraints of the ECNs, the ECNs may post portion of a given work to just the CC. In other words, the CC was seen as the greatest pool of funds to most ECNs. Consequently, a distribution of VM resources between many centralised ECNs has become a particular numerous 0-1 knapsack issue. Particularly while heterogeneous time limits are involved, that very VM allocation of resources issue throughout MEC was indeed non-trivial.

Second, their distribution of ECNs is heterogeneous. Thus, the capacity of its ECN comprises the capacity shared also with CC as well as the spectrum reserved for network devices. It suggests that even the transfer disruptions throughout the embed of certain time-sensitive activities across separate ECNs were heterogeneous. Often, when the work was transmitted mostly from ECN only to CC, the transfer interval includes two parts: the time when the work was transferred from the recipient to the ECN and the time when the work becomes submitted from its ECN to the CC. The latency in the delivery of its assignment is thus unpredictable, that relies mostly on particular resource allocation.

Third, considering the dynamic setup of ECN services, like VM resource capability, latency, connectivity costs, and so on., smartphone users will get various needs to every ECN. Mostly as consequence, different individuals can especially those with limited assets from the same ECN. Users that defeat the rivalry must pursue alternate options from several other ECNs, that lead throughout rivalry for services in other ECNs. In addition , in order to make sure where each phone node does not exploit the bids (i.e. truthfulness) or whether the payout of every other user was non-negative (i.e. human rationality), the suitable auction scheme is essential for this VM resource rivalry through several geo-distributed ECNs.

II. RELATED WORK

Throughout this article, they concentrate mostly on auction-based Vm placement issue of time-sensitive activities throughout the decentralized edge cloud. Since then, the lot of research is being conducted regarding edge cloud computing.

The researchers for [20, 36] and [23, 34] provided veritable marketplace frameworks of competitive resource scheduling as well as distribution throughout cloud computing. The writers of its preceding have taken into account heterogeneous consumer demands, although the latter have planned hybrid auction processes. In addition , the authors throughout [12, 39] suggested auctioning strategies for decentralised VM provisioning but also billing through various geo-distributed data centres. Work[16, 37] looked towards online marketplace structures to cloud infrastructure-as-a-service (IaaS) clouds, throughout which[37] concentrated on optimising both social benefit as well as advantage of vendors, while[16] found the distinctive characteristics about a flexible system through plugging time-varying customer requirements as well as a cohesive framework of finding heterogeneous VMs together. Zhang et al.[38] have developed an effective randomised auction system for complex VM provisioning as well as pricing with (1 − 5-007)-optimal social welfare throughout anticipation. The developers in[35] proposed a system for true auction sites in cloud computing where consumers of heterogeneous requests will enter and leave on the fly. In addition, Anisetti et al.[5] designed the multi-cloud supply situation only as generalisation of second price acquisition e-auctions wherein the cloud service users as well as suppliers are treated as auctioneers as well as bidders, accordingly. We suggested an e-auction policy based on pairing but also rating techniques for increase the precision of the water production. Zheng et al.[40] suggested this first group for method-proof dual transactions of multi-user, multi-purpose network reservations that could offer strategy-proofness, ex-post budget flexibility, decent social security, perfect server bandwidth usage as well as a high tenant fulfillment ratio around the same time. Unlike the current marketplace structures, they are developing the veritable auction-based VM allocation policies to time-sensitive activities throughout dispersed ECNs, where there would be an asset rivalry in several geographically scattered ECNs. In addition, the time limit but also capacity issues were also considered in this study at about the same time.

Works [14, 24, 39], but at the other hand , focused on adjusting a work pressure among different network cloud servers to reduce operational responsiveness. Other than that, some other works [12, 13, 17] investigated how to dynamically configure edge clouds through suggesting online resource positioning techniques. In addition,[25] modelled the edge cloud mostly as tree structure for geo-distributed servers as well as suggested a workload placement algorithm that maximises the number of peak workloads. [8] examined multi-user data unloading problems for mobile edge clouds throughout a multi-channel wireless interference environment. Unlike some studies, that primarily research workload scheduling and VM positioning problems throughout edge clouds, they concentrate on VM allocation with an investment point of view.

Activities that would be most important towards everyones dilemma were [15, 22, 30]. The researchers in[15] developed two dual bidding frameworks for encourage cloudlets for service neighbouring cell phones, wherein the research articles presume which demands from smartphone subscribers as well as cloudlet services were homogeneous, so that each cloudlet may support just one request. In[30], some results suggest two auction structures for two task models, that becomes basically a crowdsourcing-based channel assignment issue. Nor the analysis analyses the workarounds among the suggested pathways. The researchers throughout [22] built an interactive insurance scheme of virtual machines server providing (such as VM location including inter-VM traffic routing) but also geo-distributed cloud selling, modelled as both an online mixture knapsack issue.

From the other hand, we suggest a veritable auction-based VM wealth distribution system for time-sensitive activities throughout the decentralized edge cloud, that addresses the issue of phone devices vying with VM resources throughout widely distant ECNs with bandwidth constraints. They design the issue as that of an n-to-one weighted graph structure following the 0-1 knapsack restriction issue. In addition, they evaluate the workarounds of its winning selection method as well as further illustrate the importance of the variable. Indeed, their adverse outcomes by the most latest studies mostly on digital linear constraints as well as the planned fractal dimension mixing problem[27], wherein the previously optimization online true algorithms will time approach ratios of 2e and 24, accordingly. Besides that, the bidding approach followed throughout this report varies from either the four standard forms of auction in which the essential compensation law proposed through Myerson was applied[19]. For the meantime, the Vickrey-Clarke-Groves (VCG) auctions[26], predicated on the maximum utilization, should not be used in this paper so the n-to-one have the fitted issue with 0-1 knapsack restrictions should never be resolved efficiently. Mostly as non-true multi-item auction system, the Generalized Second-Price auction[9] was n’t suited to just the VM capital allocation issue that wants to maintain authenticity.

III. METHODOLOGY

AUCTION-BASED VM ALLOCATION

Throughout this segment, they suggest the Auction-based VM resource Allocation (AVA) framework for time-sensitive tasks in geo-distributed edge clouds. The AVA process largely consists of the Winning Bid Collection (WBS) algorithm and the Reward Confirmation (PD) method. Firstly, they evaluate the NP-hardness of WBS problem, instead they develop the greedy winning bid selection technique as well as a true payment selection method for overcome the WBS as well as PD issues, separately.

Problem Hardness Analysis

First of all, they demonstrate also that WBS can't be solved throughout polynomial time when P = NP. In fact, they get the following theorem:

Theorem 1. 1. The issue with WBS was NP-hard. Clear evidence: firstly, we find the unique circumstance of its WBS issue, where even the ECNs could not submit duties to just the CC, the amount for ECNs becomes close to unity, but at the same period we allow Ti ≥di1 = Ii / b+1 to upload activities to the CC, the amount of ECNs is equal to 1. Presently, we 're adding the trivial 0-1 knapsack problem [28]: "maximise: valuen i=1 wi •xi, subject to: valuen i=1 vi •xi ≤ C, full value {0 , 1}." Here, wi and vi signify the size and weight of its I object, as well as C indicate the size of its knapsack.

Through assigning C, wi as well as vi throughout the trivial 0-1 knapsack problem for L1, vi1−Ai •c v 1 but also Ai throughout the particular WBS case, they have the significant issues to be of the same. In other words, the unique case of its WBS major issue seems to be a trivial 0-1 knapsack issue, which would be NP-hard. Its most large issue with WBS was thus at least NP-hard.

Winning Bid Selection:

Specific Answer The WBS dilemma here is to pick a final bid such that we all can optimise positive value whereas guaranteeing that even the time limit for proposals as well as the power limitations of the ECNs will be met at the same time. Since the WBS dilemma does have time limits but also network conditions, we are splitting our answer into following stages. In several steps, we take this into account time limits as well as storage capacity.

First phase: We focus on removing the deadline constraints of requests. First, we use d 1 ij = Ii/b↓ j and d 2 ij =Ii/b↓ j +Ii/b↑ j to denote the transmission delay of uploading ri to sj and the transmission delay of uploading ri to the CC through sj , respectively. Apparently, we have d 2 ij ≥ d 1 ij . For simplicity, we call d 1 ij and d 2 ij the good and bad transmission delay, respectively. Second, according to the relationships of Ti , d 1 ij and d 2 ij , we update the bid set B and the ECN set S. That is, we will remove the bids which cannot satisfy the deadline constraints, and add some virtual bids and ECNs if the deadline is larger than the bad transmission delay. More specifically, we have the three following cases:

- Case 1: when the ri deadline was shorter than that of the decent propagation interval, i.e. Ti < d1 ij, they can exclude the bid bij immediately from its B bid. It was because the bid bij could n't reach the time limit.
- Case 2: when Ti is within a poor queuing delay but really not just over a decent queuing delay, i.e. d 1 ij ≤Ti < d 2 ij ij, they would not make any decision.
- Case 3: whether Ti is not less than that of the incorrect delay time, i.e., Ti ≥ d 2 ij, they can build the digital ECN sj function as well as a digital bid bij rán, while sj kan = {Lj kan = Ai, cv j kan = ct j+c v 0, ct j kan = b j kan = b j kan = 0} as well as the worth of bij kan is equal to that of bij kan. Here, the method of creating digital ECNs but also bids includes extracting the CC as from edge cloud computing scenario and converting the three-layer edge cloud system into a two-layer cloud structure. S as well as B are used to signify virtual ECN but also bid sets, however. Now we will connect S and B , respectively, to virtual ECN sjVirtual as well as the virtual bid bij Virtual.

Second phase: We 're focused mostly on WBS issue of power restrictions. They initially design the highest offer collection as more of a weighed n-to-one graph structure fitting the 0-1 knapsack restriction issue. Because the dilemma was NP-hard due to the power limitations, they are following the selfish model to build the optimal pairing, but has an estimated limit actual volume. To convenience, let Sb but also Bb denote modified ECN and bid sets containing digital ECNs as well as bids, respectively. After all, Sb = S with S and Bb = B with B. The successful bids are therefore computed as follows.

Initially, they create the weighed bi-part graph containing power restrictions, indicated as G={R, Sb, E: B}b, where Rand Sb were 2 distinct vertex sets, whereas E corresponds to the edge set between R and Sb. The edge between ri and sj is denoted by us as a comfort. In this case, each bid by Bb refers to the edge by, sj by. At each vertex, the necessary VM resources Ai is seen as the volume of the item in the trivial 0-1 knapsack, whereas at each vertex, a VM network resource Lj should be seen as the volume of its knapsack. In addition, every other edge of the SJA correlates towards a weighted specified to welfare benefits for element of VM property. Let wij indicate the strength of it's corner ⟨ri ,sj⟩∈ E . Next, we've got

wij = bij Ai − c v j , for ∀⟨ri , sj⟩∈ E. (9)

Algorithm 1 Preprocessing Algorithm

Require: R, S, B, c v 0 .

Ensure: S, B, Sb, and Bb.

1: Initialize S = B = ϕ;
2: for ri∈ R do
3: for sj∈ S do
4: d 1 ij = Ii/b↓ j , d 2 ij = Ii/b↓ j + Ii/b↑ j ;
5: if Ti < d1 ij then
6: Remove bij from B, i.e., B = B − {bij};
7: else if d 1 ij ≤ Ti < d2 ij then
8: Continue; //Case 2;
9: else if d 2 ij ≤ Ti then
10: Generate a virtual ECN sj∗, S = S + {sj∗}; //sj∗={Lj∗ =Ai , cv j∗ =c t j+c v 0 , ct j∗ =b ↑ j∗ =b ↓ j∗ = 0};
11: Generate a virtual bid bij∗, B = B + {bij∗};
12: Sb = S∪S and Bb = B∪B;
13: return S, B, Sb and Bb;

Dependent mostly on weighted bipartite given Graph, the WBS issue are being generalised as well as re-formalised. While deleting the CC as well as inserting few digital ECNs including bids, the three-layer edge cloud configuration was modified to just a two-layer configuration. Again for two-layer edge cloud layout, only one ECN can be assigned per each domain. Then we'll get the highest offer fixed to C = Δ, but we'll have the highest offer set to name = ΔE. They are therefore reformulating the WBS issue:

max ∑ bij∈Φ(bij−Ai · c v j) (10)
s.t. Φ ⊆ Bb, (11)
∑ j:bij∈Φ 1 ≤ 1, ∀ri∈ R (12)
∑ i:bij∈Φ Ai ≤ Lj , ∀sj∈Sb (13)
Second, while creating a weighed bipartite graph and power restrictions, they eagerly pick those corners for shape the perfect fit for G with an arbitrary weight limit. Most precisely, we pick the end only with greatest weight in-ring. Without a lack for generalisation, just let edge be toll-free. Currently, we've received the correct two strategies
(1) Ai ≤Lj: it implies also that available potential Lj of ECN sj was n't worse than the sum of Ai capital. Then we apply the appropriate bid bij to just the task solution. In addition, we explicitly detach the request vertex ri from R and delete all of the related ri outlines to E in G. They also affect the price of Lj by removing Ai from the vertex sj
(2) Ai > Lj: we remove the edge ⟨ri , sj⟩ from Eand continue to find the edge with the next largest weight
Notice that its corresponding mass also was lost whenever the edge was lost from of the bipartite graph. Such collection cycle is continued unless R, Sb, or E is now an absolute value. Eventually, we 're going to have a task solution. Unless the bid was correct, it means that now the bid bij wins as well as the offer ri is allocated to the ECN sj.
Since the highest bid has been finalised, they instead create planning choices at each bid. Such that, for highest offer bij, they have to let sj finish the mission ri rather than submit the ri to just the CC via sj, i.e., they have to decide the winning bid schedule of the expected to win bids set by E but also C.
Depending from the above response, they initially develop the pre - processing Method, such as in Algorithm 1, for eliminate time limits. Through removing the number of bids as well as inserting few synthetic bids including ECNs, Algorithm 1 will change the ECN set and the bid collection. Most precisely, throughout Step 1, we initialise the synthetic ECN or the bid collection (i.e., S and B). The optimistic but its also weak transmission delay (i.e., d 1 ij but also d 2 ij) is calculated in Phases 2-4. Similar to just the interaction between Ti, d 1 ij as well as d 2 ij ij, we are withdrawing a few of the bids which could not meet the time limits set out during Phases 5-6. Throughout Phases 7-8, if d 1 ij ≤ Ti < d2 ij, they do not take any action. When d 2 ij ≤ Ti ~, they initially create a different ECN sj License as well as a digital bid bij License but instead apply it to S but B , respectively, throughout Phases 9-11. In Stage 12, we'll have the modified ECN set Sb as well as the new bid set Bb. Finally, Algorithm 1 outputs S, B, Sb as well as Bb throughout phase 13.
Next, we create the calibrated two-part graph to storage limitations G = {R, Sb, E: B}b. Premised upon it, we are designing an avaricious Winning Bid Selection (WBS) method shown in figure 1. Initially, we will initialise the task solution throughout phase 1. Then we're going through the greedy winning bid selection procedure in Steps 2-9. Explicitly, throughout phase three, the layer (e.g., sj) with largest weight (i.e., wij) has been picked. After which we try comparing the remaining Lj capacity of SJ only with amount of Ai VM resources in Step 4. If Ai ≤Lj, the assessment remedy will be extended to Step 5, i.e. to name = name+{bij}. Throughout phases 6-7, most corners associated with the vertex ri throughout G were also removed as well as the vertex mark Lj has been modified by deducting Ai. When Ai > Lj . When R, Sb, or E is now an original image, Algorithm 2 ends but also outcomes throughout Step 10.

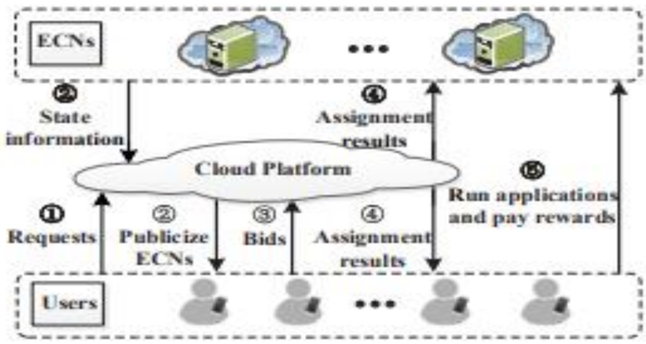


Fig: The auction model for distributed Edge cloud

Algorithm 3
Winning Bid Scheduling Algorithm
Require: Φ and S. Ensure: Φ E and Φ C .
1: Initialize Φ E = Φ C = ϕ;
2: for bij∈ Φ (corresponding to ⟨ri ,sj⟩) do
3: if sj∈ S then
4: Φ C = ΦC +{bij};
5: else
6: Φ E = ΦE +{bij};
7: return Φ E and Φ C ;
3.4 Payment Determination:
Specific Approach The real price calculation determines the cost to every successful bid, meaning how every customer is realistic about their true value of their Cloud Service order. Throughout this article, they follow the essential payment rule proposed through Myerson[19] to decide the reward for each winning bid.
Algorithm 4
Payment Determination (PD) Algorithm
Require: G = {R, Sb, E :Bb}, Φ = ΦE ∪Φ C and B.
Ensure: P = {pij (bij)|bij∈Φ}.
1: for each bij∈Φ do
2: E−ij =E−{⟨ri , sj⟩} and G−ij ={R, Sb, E−ij :B−{bij}};
3: Execute Algorithm 2: Φ−ij = WBS(G−ij);
4: Determine wiminj = min{wi1j , wi2j , · · · , wiкj∈ Φ−ij : Lj− ∑ wixj≥wiyj Aix ≥Ai};
5: pij (bij)=Ai ·(c v j +max{wij′ , wiminj}); // bij′ ∈Φ−ij

6: if bij∈ B then

7: User ri pays the ECN sj∗ with the reward pij (bij); //sj here is the virtualization of sj∗, i.e., //sj ={Lj =Ai , cv j =c t j∗+c v 0 , ct j =b ↑ j =b ↓ j = 0};

8: else

9: User ri pays the ECN sj with the reward pij (bij); // sj here is a real ECN.

10: return P;

Mostly on basis of the other procedure, the transaction decision method has been seen in Algorithm 4. By each successful bid bij throughout the initial solution (i.e., bij) they initially recreate the current bipartite graph G−ij lacking bij throughout phase two. And we'll run the biased WBS model called on G−ij input and maybe get a new assignment solution named name−ij in Step 3. There, they then use type "Output = WBS(Input)" for represent the WBS method to simplicity. In Step 4, we evaluate the essential mass of the ECN sj, i.e. wiminj, as per Eq. 14. 14. The crucial compensation for bid bij is decided as per Eq in Phase 5. 15. Whether Bij would be a digital bid, we will define the ECN that will company can pay throughout Phases 6-7. This would be to say, the actual ECN sj file whose sj seems to be the virtual machine of sj file, i.e. ,sj = {Lj = Ai, cv j = c t j file+c v 0, ct j = b file j = b file j = 0}, would be paying pij (bij). Anything else, if Bij would be a valid bid, i.e. Bij-B, the valid ECN sj would automatically collect the payment pij (bij) throughout Phases 8-9. Upon calculating vital transactions with all successful bids, its method ends as well as extracts the findings throughout Step 10.

IV. RESULTS AND DISCUSSION

1. When a mobile user wishes to rent VM resources to run its deadline-sensitive cloud computing applications, it first generates a request and then submits the request to the platform. The request is composed of the user's maximum tolerable latency (i.e., deadline), the amount of required VM resources and the amount of input data. We use $r_i = T_i, A_i, I_i$ to denote the i-th mobile user's request, where T_i, A_i and I_i mean the deadline, the total VM resources and the amount of input data. Moreover, the set of all requests is denoted by R.

2. The platform will periodically collect the state in- formation of each ECN, and then publicize it to the mo- bile users who submit requests. The state information includes several main parameters: VM resource capacity, bandwidths, unit cost of renting VM resources and unit cost of transmitting data to the CC.

3. Then, the mobile user values differently to the ECNs according to the state information. At the same time, the mobile user determines a bid for each ECN. For each request r_i, we use b_{ij} and v_{ij} to denote the bid B and valuation of i-th mobile user to the j-th ECN, respectively

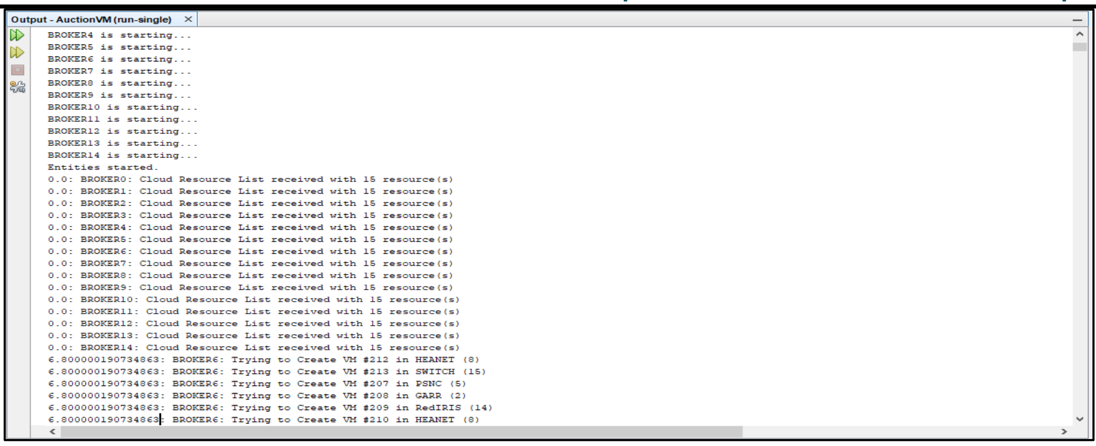
4. .Based on the received bids and requests from mobile users, the platform determines the winners of the auction, makes the scheduling decision for the winners, and computes the corresponding payments.

5. Mobile users upload the input data to the ECNs to run their cloud computing applications, and then pay the corresponding rewards. According to the scheduling decisions for winning bids, ECNs decide to upload some tasks to the CC and pay the CC.

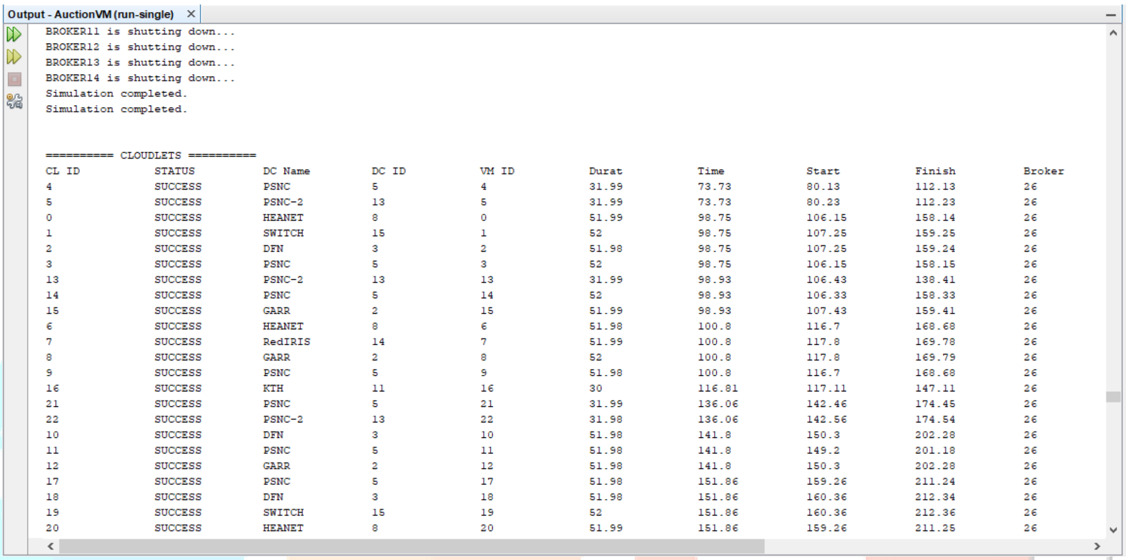
The winning bidding selection problem:

The WBS problem is how to select the winning bid set, so that we can maximize the social welfare, while ensuring that the deadline constraints of requests and the capacity constraints of ECNs can be satisfied simultaneously. Since the WBS problem has both deadline constraints and capacity constraints, we divide our solution into two phases. We take the deadline and capacity constraints into consideration in the two phases, respectively

1. Deadline constraint removal
2. Capacity constraints removal


Output - AuctionVM (run-single)

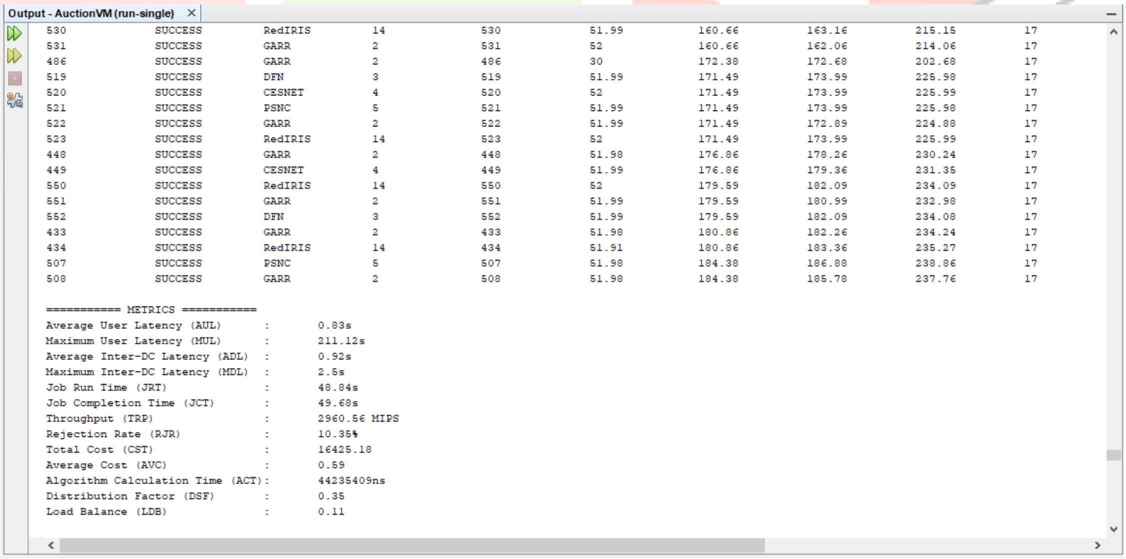
```
BROKER4 is starting...
BROKER5 is starting...
BROKER6 is starting...
BROKER7 is starting...
BROKER8 is starting...
BROKER9 is starting...
BROKER10 is starting...
BROKER11 is starting...
BROKER12 is starting...
BROKER13 is starting...
BROKER14 is starting...
Entities started.
0.0: BROKER0: Cloud Resource List received with 15 resource(s)
0.0: BROKER1: Cloud Resource List received with 15 resource(s)
0.0: BROKER2: Cloud Resource List received with 15 resource(s)
0.0: BROKER3: Cloud Resource List received with 15 resource(s)
0.0: BROKER4: Cloud Resource List received with 15 resource(s)
0.0: BROKER5: Cloud Resource List received with 15 resource(s)
0.0: BROKER6: Cloud Resource List received with 15 resource(s)
0.0: BROKER7: Cloud Resource List received with 15 resource(s)
0.0: BROKER8: Cloud Resource List received with 15 resource(s)
0.0: BROKER9: Cloud Resource List received with 15 resource(s)
0.0: BROKER10: Cloud Resource List received with 15 resource(s)
0.0: BROKER11: Cloud Resource List received with 15 resource(s)
0.0: BROKER12: Cloud Resource List received with 15 resource(s)
0.0: BROKER13: Cloud Resource List received with 15 resource(s)
0.0: BROKER14: Cloud Resource List received with 15 resource(s)
6.800000190734863: BROKER6: Trying to Create VM #212 in HEANET (8)
6.800000190734863: BROKER6: Trying to Create VM #213 in SWITCH (15)
6.800000190734863: BROKER6: Trying to Create VM #207 in PSNC (5)
6.800000190734863: BROKER6: Trying to Create VM #208 in GARR (2)
6.800000190734863: BROKER6: Trying to Create VM #209 in RedIRIS (14)
6.800000190734863: BROKER6: Trying to Create VM #210 in HEANET (8)
```


Output - AuctionVM (run-single)

```
BROKER11 is shutting down...
BROKER12 is shutting down...
BROKER13 is shutting down...
BROKER14 is shutting down...
Simulation completed.
Simulation completed.
```

========== CLOUDLETS ==========

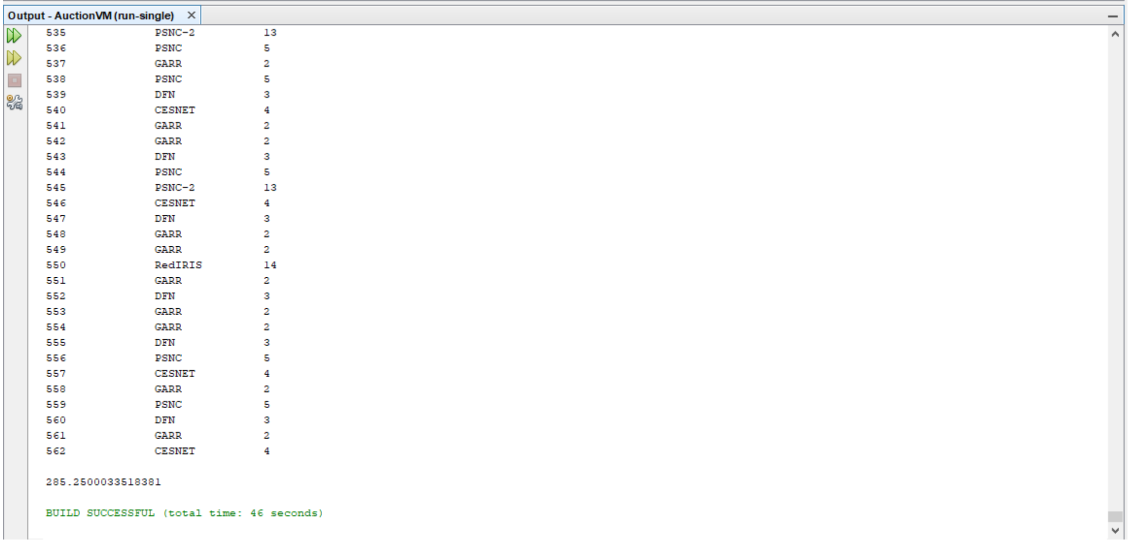
CL ID	STATUS	DC Name	DC ID	VM ID	Durat	Time	Start	Finish	Broker
4	SUCCESS	PSNC	5	4	31.99	73.73	80.13	112.13	26
5	SUCCESS	PSNC-2	13	5	31.99	73.73	80.23	112.23	26
0	SUCCESS	HEANET	8	0	51.99	98.75	106.15	158.14	26
1	SUCCESS	SWITCH	15	1	52	98.75	107.25	159.25	26
2	SUCCESS	DFN	3	2	51.98	98.75	107.25	159.24	26
3	SUCCESS	PSNC	5	3	52	98.75	106.15	158.15	26
13	SUCCESS	PSNC-2	13	13	31.99	98.93	106.43	138.41	26
14	SUCCESS	PSNC	5	14	52	98.93	106.33	158.33	26
15	SUCCESS	GARR	2	15	51.99	98.93	107.43	159.41	26
6	SUCCESS	HEANET	8	6	51.98	100.8	116.7	168.68	26
7	SUCCESS	RedIRIS	14	7	51.99	100.8	117.8	169.78	26
8	SUCCESS	GARR	2	8	52	100.8	117.8	169.79	26
9	SUCCESS	PSNC	5	9	51.98	100.8	116.7	168.68	26
16	SUCCESS	KTH	11	16	30	116.81	117.11	147.11	26
21	SUCCESS	PSNC	5	21	31.99	136.06	142.46	174.45	26
22	SUCCESS	PSNC-2	13	22	31.98	136.06	142.56	174.54	26
10	SUCCESS	DFN	3	10	51.98	141.8	150.3	202.28	26
11	SUCCESS	PSNC	5	11	51.98	141.8	149.2	201.18	26
12	SUCCESS	GARR	2	12	51.98	141.8	150.3	202.28	26
17	SUCCESS	PSNC	5	17	51.98	151.86	159.26	211.24	26
18	SUCCESS	DFN	3	18	51.98	151.86	160.36	212.34	26
19	SUCCESS	SWITCH	15	19	52	151.86	160.36	212.36	26
20	SUCCESS	HEANET	8	20	51.99	151.86	159.26	211.25	26

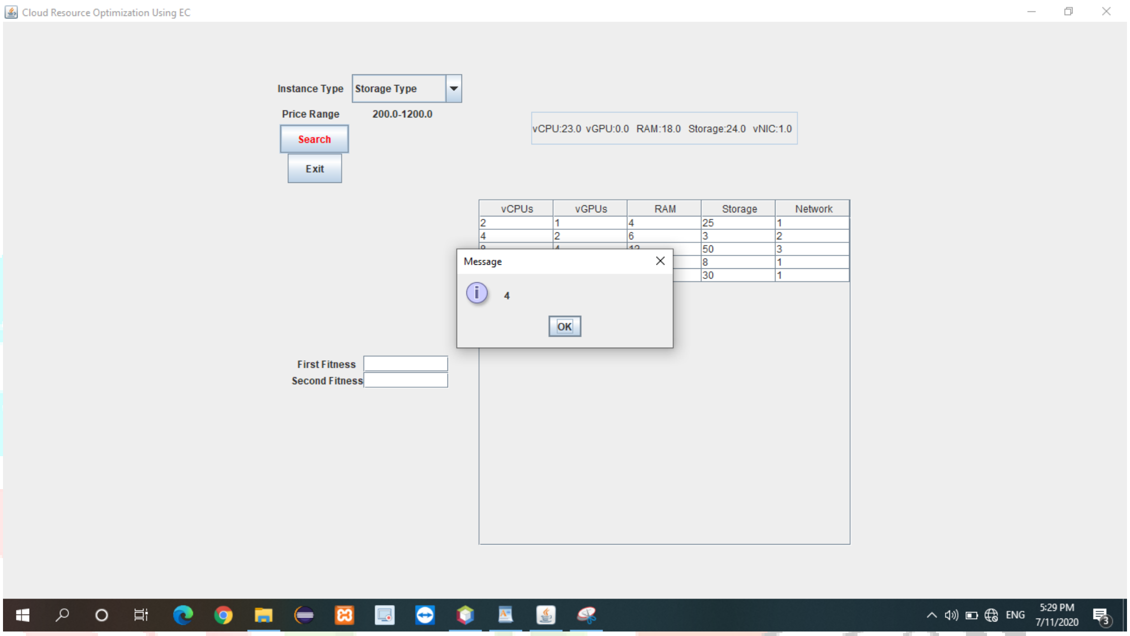

Output - AuctionVM (run-single)

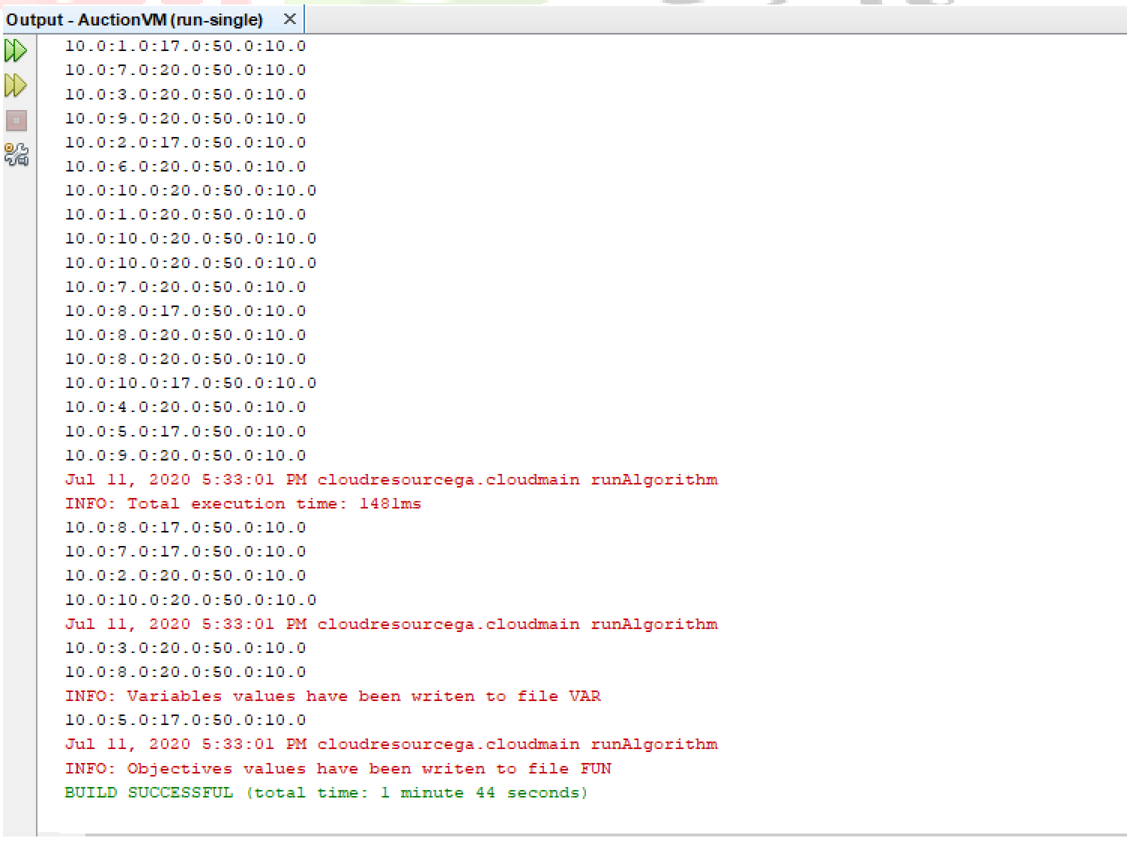
530	SUCCESS	RedIRIS	14	530	51.99	160.66	163.16	215.15	17
531	SUCCESS	GARR	2	531	52	160.66	162.06	214.06	17
486	SUCCESS	GARR	2	486	30	172.38	172.68	202.68	17
519	SUCCESS	DFN	3	519	51.99	171.49	173.99	225.98	17
520	SUCCESS	CESNET	4	520	52	171.49	173.99	225.99	17
521	SUCCESS	PSNC	5	521	51.99	171.49	173.99	225.98	17
522	SUCCESS	GARR	2	522	51.99	171.49	172.89	224.88	17
523	SUCCESS	RedIRIS	14	523	52	171.49	173.99	225.99	17
448	SUCCESS	GARR	2	448	51.98	176.86	178.26	230.24	17
449	SUCCESS	CESNET	4	449	51.99	176.86	179.36	231.35	17
550	SUCCESS	RedIRIS	14	550	52	179.59	182.09	234.09	17
551	SUCCESS	GARR	2	551	51.99	179.59	180.99	232.98	17
552	SUCCESS	DFN	3	552	51.99	179.59	182.09	234.08	17
433	SUCCESS	GARR	2	433	51.98	180.86	182.26	234.24	17
434	SUCCESS	RedIRIS	14	434	51.91	180.86	183.36	235.27	17
507	SUCCESS	PSNC	5	507	51.98	184.38	186.88	238.86	17
508	SUCCESS	GARR	2	508	51.98	184.38	185.78	237.76	17

=========== METRICS ===========

```
Average User Latency (AUL)          :      0.83s
Maximum User Latency (MUL)          :      211.12s
Average Inter-DC Latency (ADL)      :      0.92s
Maximum Inter-DC Latency (MDL)      :      2.5s
Job Run Time (JRT)                  :      48.84s
Job Completion Time (JCT)           :      49.68s
Throughput (TRP)                    :      2960.56 MIPS
Rejection Rate (RJR)                :      10.35%
Total Cost (CST)                    :      16425.18
Average Cost (AVC)                  :      0.59
Algorithm Calculation Time (ACT)    :      44235409ns
Distribution Factor (DSF)           :      0.35
Load Balance (LDB)                  :      0.11
```


Output - AuctionVM (run-single)
535 PSNC-2 13
536 PSNC 5
537 GARR 2
538 PSNC 5
539 DFN 3
540 CESNET 4
541 GARR 2
542 GARR 2
543 DFN 3
544 PSNC 5
545 PSNC-2 13
546 CESNET 4
547 DFN 3
548 GARR 2
549 GARR 2
550 RedIRIS 14
551 GARR 2
552 DFN 3
553 GARR 2
554 GARR 2
555 DFN 3
556 PSNC 5
557 CESNET 4
558 GARR 2
559 PSNC 5
560 DFN 3
561 GARR 2
562 CESNET 4
285.2500033518381
BUILD SUCCESSFUL (total time: 46 seconds)


Cloud Resource Optimization Using EC
Instance Type
Storage Type
Price Range
200.0-1200.0
Search
Exit
vCPU:23.0 vGPU:0.0 RAM:18.0 Storage:24.0 vNIC:1.0
vCPUs vGPUs RAM Storage Network
Message
4
OK
First Fitness
Second Fitness
ENG
5:29 PM
7/11/2020


Output - AuctionVM (run-single)
10.0:1.0:17.0:50.0:10.0
10.0:7.0:20.0:50.0:10.0
10.0:3.0:20.0:50.0:10.0
10.0:9.0:20.0:50.0:10.0
10.0:2.0:17.0:50.0:10.0
10.0:6.0:20.0:50.0:10.0
10.0:10.0:20.0:50.0:10.0
10.0:1.0:20.0:50.0:10.0
10.0:10.0:20.0:50.0:10.0
10.0:10.0:20.0:50.0:10.0
10.0:7.0:20.0:50.0:10.0
10.0:8.0:17.0:50.0:10.0
10.0:8.0:20.0:50.0:10.0
10.0:8.0:20.0:50.0:10.0
10.0:10.0:17.0:50.0:10.0
10.0:4.0:20.0:50.0:10.0
10.0:5.0:17.0:50.0:10.0
10.0:9.0:20.0:50.0:10.0
Jul 11, 2020 5:33:01 PM cloudresourcega.cloudmain runAlgorithm
INFO: Total execution time: 1481ms
10.0:8.0:17.0:50.0:10.0
10.0:7.0:17.0:50.0:10.0
10.0:2.0:20.0:50.0:10.0
10.0:10.0:20.0:50.0:10.0
Jul 11, 2020 5:33:01 PM cloudresourcega.cloudmain runAlgorithm
10.0:3.0:20.0:50.0:10.0
10.0:8.0:20.0:50.0:10.0
INFO: Variables values have been writen to file VAR
10.0:5.0:17.0:50.0:10.0
Jul 11, 2020 5:33:01 PM cloudresourcega.cloudmain runAlgorithm
INFO: Objectives values have been writen to file FUN
BUILD SUCCESSFUL (total time: 1 minute 44 seconds)

V. CONCLUSION

Throughout this article, they dealt with the issue of assigning heterogeneous VM capacity requirements for geo-distributed edge cloud nodes of bandwidth limitations for optimise overall welfare programs. They suggest the veritable auction-based VM wealth distribution system , i.e. AVA, consisting often of a selfish highest offer collection method as well as a veritable compensation selection method. They have seen that the AVA system can maintain the values for authenticity, human rationality as well as computationally efficient. In addition, they offer a redundancies of its winning prediction method as well as demonstrate this to be an immediate required. Eventually, detailed experiments focused on actual trace validate the output of the AVA system. Throughout ongoing studies, the issue of VM allocating resources for interactive cloud computing involving transfer with cloud services tasks would be investigated.

REFERENCES

[1] Grid workloads archive. Available [Online]: http://gwa.ewi.tudelft.nl, Aug. 2018.
[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1):450–465, 2018.
[3] A. Aissioui, A. Ksentini, A. Gueroui, and T. Taleb. On enabling 5g automotive systems using follow me edgecloud concept. IEEE Transactions on Vehicular Technology, 67(6):5302–5316, 2018.
[4] Amazon. “Amazon Elastic Compute Cloud”. Available [Online]: https://aws.amazon.com/cn/ec2/, Aug. 2018.
[5] M. Anisetti, C. A. Ardagna, P. A. Bonatti, E. Damiani, M. Faella, C. Galdi, and L. Sauro. E-auctions for multicloud service provisioning. In 2014 IEEE International Conference on Services Computing, pages 35–42, 2014.
[6] A. Ceselli, M. Premoli, and S. Secci. Mobile edge cloud network design optimization. IEEE/ACM Transactions on Networking, 25(3):1818–1831, 2017.
[7] L. Chen, J. Wu, X. Zhang, and G. Zhou. TARCO: Two-stage auction for d2d relay aided computation resource allocation in hetnet. IEEE Transactions on Services Computing, pages 1–1, 2018.
[8] X. Chen, L. Jiao, W. Li, and X. Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Transactions on Networking, 24(5):2795– 2808, 2016.
[9] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. American economic review, 97(1):242–259, 2007.
[10] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu. Truthful incentive mechanism for nondeterministic crowdsensing with vehicles. IEEE Transactions on Mobile Computing, 17(12):2982–2997, 2018.
[11] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu. Fogroute: Dtn-based data dissemination model in fog computing. IEEE Internet of Things Journal, 4(1):225–235, 2017.
[12] F. Hao, M. Kodialam, T. V. Lakshman, and S. Mukherjee. Online allocation of virtual machines in a distributed cloud. IEEE/ACM Transactions on Networking, 25(1):238– 249, 2017.
[13] I.-H. Hou, T. Zhao, S. Wang, and K. Chan. Asymptotically optimal algorithm for online reconfiguration of edgeclouds. In ACM MobiHoc, pages 291–300, 2016.
[14] M. Jia, W. Liang, Z. Xu, and M. Huang. Cloudlet load balancing in wireless metropolitan area networks. In IEEE INFOCOM, pages 1–9, 2016.
[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, 9(6):895–909, 2016.
[16] J. Li, Y. Zhu, J. Yu, C. Long, G. Xue, and S. Qian. Online auction for IaaS clouds: Towards elastic user demands and weighted heterogeneous VMs. In *IEEE INFOCOM*, pages 1–9, 2017.
[17] Y. Li and W. Wang. Can mobile cloudlets support mobile applications? In *IEEE INFOCOM*, pages 1060–1068, 2014.
[18] Microsoft. “Windows Azure”. Available [Online]: http://www.windowsazure.com/, Aug. 2018.
[19] R. B. Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.
[20] M. M. Nejad, L. Mashayekhy, and D. Grosu. Truthful greedy mechanisms for dynamic virtual machine provi-sioning and allocation in clouds. *IEEE Transactions on Parallel and Distributed Systems*, 26(2):594–603, 2015.
[21] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato. Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control. *IEEE Transactions on Computers*, 66(5):810–819, 2017.
[22] W. Shi, C. Wu, and Z. Li. An online mechanism for dynamic virtual cluster provisioning in geo-distributed clouds. In *IEEE INFOCOM*, pages 1–9, 2016.
[23] W. Shi, L. Zhang, C. Wu, Z. Li, and F. C. M. Lau. An online auction framework for dynamic resource provisioning in cloud computing. *IEEE/ACM Transactions on Networking*, 24(4):2060–2073, 2016.
[24] H. Tan, Z. Han, X.-Y. Li, and F. C. Lau. Online job dispatch-ing and scheduling in edge-clouds. In *IEEE INFOCOM*, pages 1–9, 2017.
[25] L. Tong, Y. Li, and W. Gao. A hierarchical edge cloud architecture for mobile computing. In *IEEE INFOCOM*, pages 1–9, 2016.
[26] H. R. Varian and C. Harris. The VCG auction in theory and practice. *American Economic Review*, 104(5):442–45,2014.
[27] R. Vaze. Online knapsack problem and budgeted truthful bipartite matching. In *IEEE INFOCOM*, pages 1–9, 2017.
[28] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
[29] H. Wang, Z. Kang, and L. Wang. Performance-aware cloud resource allocation via fitness-enabled auction. *IEEE Transactions on Parallel and Distributed Systems*, 27(4):1160– 1173, 2016.
[30] X. Wang, X. Chen, and W. Wu. Towards truthful auction mechanisms for task assignment in mobile device clouds. In *IEEE INFOCOM*, pages 1–9, 2017.
[31] X. Wang, K. Wang, S. Wu, D. Sheng, H. Jin, K. Yang, and S. Ou. Dynamic resource scheduling in mobile edge cloud with cloud radio access network. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–1, 2018.
[32] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo. Efficient algorithms for capacitated cloudlet placements. *IEEE Transactions on Parallel and Distributed Systems*, 27(10):2866– 2880, 2016.
[33] H. Yin, X. Zhang, H. H. Liu, Y. Luo, C. Tian, S. Zhao, and F. Li. Edge provisioning with flexible server place-ment. *IEEE Transactions on Parallel and Distributed Systems*, 28(4):1031–1045, 2017.
[34] S. Zaman and D. Grosu. A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in

clouds. *IEEE Transactions on Cloud Computing*, 1(2):129– 141, 2013.
[35] H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu. A framework for truthful online auctions in cloud comput-ing with heterogeneous user demands. *IEEE Transactions on Computers*, 65(3):805–818, 2016.
[36] L. Zhang, Z. Li, and C. Wu. Dynamic resource provision-ing in cloud computing: A randomized auction approach. In *IEEE INFOCOM*, pages 433–441, 2014.
[37] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau. Online auctions in IaaS clouds: Welfare and profit maximization with server costs. *IEEE/ACM Transactions on Networking*, 25(2):1034–1047, 2017.
[38] X. Zhang, C. Wu, Z. Li, and F. C. Lau. A truthful (1-″)-optimal mechanism for on-demand cloud resource provi sioning. In *IEEE INFOCOM*, pages 1053–1061, 2015.
[39] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. C. M. Lau. Dynamic pricing and profit maximization for the cloud with geo-distributed data centers. In *IEEE INFOCOM*, pages 118–126, 2014.
[40] Z. Zheng, Y. Gui, F. Wu, and G. Chen. Star: Strategy-proof double auctions for multi-cloud, multi-tenant bandwidth reservation. *IEEE Transactions on Computers*, 64(7):2071– 2083, 2015.
[41] Z. Zhou, F. Liu, Z. Li, and H. Jin. When smart grid meets geo-distributed cloud: An auction approach to datacenter demand response. In *IEEE INFOCOM*, pages 2650–2658, 2015.