Optimizing Social Welfare for Task Offloading in Mobile Edge Computing

Hsiang-Jen Hong Wenjun Fan C. Edward Chow Xiaobo Zhou Sang-Yoon Chang
University of Colorado Colorado Springs, Colorado Springs, United States, CO 80918
{hhong, wfan, cchow, xzhou, schang2}@uccs.edu

***Abstract*—While mobile applications are increasing in use and complexity, the computational constraints on mobile devices remain as the bottleneck for serving computation-intensive mobile applications. Mobile edge computing (MEC) provides a computing paradigm to serve the computational demands of such mobile applications by offloading the mobile devices' computational tasks to the edge servers. Double auction has been adopted in MEC to provide a mechanism to assign the tasks of mobile devices to the edge servers while considering the satisfaction level for both entities. We improve the double auction mechanism beyond prior research in MEC. Specifically, we construct a model to support the real-world practices in the pricing scheme of edge computing, such as that provided by Amazon, and to support the parallelizing and distributing of workloads to multiple edge servers. We propose an efficient mechanism to achieve the optimal social welfare by converting the allocation problem to a minimum cost flow problem. In addition to reaching the optimal social welfare in polynomial time computations, our proposed mechanism achieves individual rationality and strong balance budget.**

***Index Terms*—Double auction, mobile edge computing, network economics, social welfare, resource allocation**

I. Introduction

As mobile applications increase in their use and functionalities, the finite constraints on the mobile devices' computations and energy become the bottleneck for such application processing. To address such issue, Mobile Edge Computing (MEC) [1] provides a capable computing paradigm to offload the computationally intensive application loads to the edge servers as opposed to processing them on the local devices. MEC provides cloud-computing capabilities and an IT service environment at the edge of the mobile networks. MEC has received increased attention in recent years and is envisioned to be a critical technique in 5G network due to its desirable properties such as low latency, proximity, high bandwidth, Specifically, mobile devices can offload some computational tasks to its adjacent edge servers to obtain a quicker response for getting the corresponding results.

In MEC, how to manipulate the task offloading is a critical issue. Specifically, edge service providers want to maximize their profits yielded by providing computational resources. Mobile devices want to save their payment and get the computational resources. In this regard, network economics and pricing models should be explicitly addressed. Auction mechanisms are popular trading schemes for dealing with the issue in MEC, a double auction mechanism especially. In a double auction, mobile devices (buyers) request to offload their computational tasks with claimed bids. Edge service providers (sellers) provide their computing resources with ask prices. An auctioneer collects those information and decides the finalized assignment and prices for them to clear the market. A double auction approach is superior to other auction approaches because the ask price can guarantee the edge service providers reach a satisfactory level in profits. In contrast, in a simpler approach, such as pay-as-bid, the buyer can lower their bidding values and decrease the profits for the edge service providers.

Our approach distinguishes itself from prior work by using a model which is more realistic and flexible. Our model supports the following three aspects not captured in the previous research. First, our model supports the current real-world practices for MEC, such as Amazon Lambda@Edge [2], by having the price proportional to the amount of load (providing finer granularity and sophistication than the prior work). Second, our model supports the many-to-many assignment, including the case of one mobile device load splitting to many edge servers; prior literature in contrast only considers the many-to-one case where one edge server services multiple mobile devices. By splitting workloads to edge servers, the tasks can be distributedly processed and the resources on the edge servers can be best utilized. We use a motivating example in section III to show that our mechanism supports this distributed case. More importantly, in order to satisfy economic efficiency objectives (measured by social welfare in our approach), such distributed concept is needed for reaching the optimality. Third, our model supports the heterogeneity of the edge servers in their service prices and qualities. Hence, our model is different from the traditional double auction model where the sellers offer identical products. The different edge servers bring the combinatorial issue. Therefore, the traditional double auction mechanism such as the breakeven approach utilized in previous literature, e.g., [3], [4], is not suitable in our model. Specifically, it is much more challenging to design a mechanism satisfying the desirable properties considered in a double auction and reaching the optimality.

From our model, we convert the allocation problem to a minimum cost flow problem and adopt the efficient cost-scaling push-relabel algorithm [5] for solving the minimum cost flow problem. Our work focuses on modeling (enabling such algorithm), applying, and analyzing the algorithm in the context of MEC (the use of such an algorithm is new in MEC and provides superior performances). Given the bidding and the ask prices, we discover that there is a flexible range of values on pricing for optimizing social welfare. The main contributions are summarized as follows. We focus on optimizing the social welfare in double auction mechanism design for

Annex to ISBN 978-3-903176-28-7 © 2020 IFIP

a task offloading problem in MEC. The proportional pricing and the many-to-many allocation better models the real-world practice in MEC, e.g., Amazon, than the previous research. Enabled by our model, we reduce the problem to the minimum cost flow model. By utilizing a state-of-the-art algorithm [5], we can obtain the best offloading decision with optimal social welfare while achieving the economic properties of individual rationality and strong balanced budget.

II. Problem Formulation

A. System Model

We define the task offloading problem using a double auction mechanism in MEC as follows. Given a set of n mobile devices $MD = \{md_i | 1 \leq i \leq n\}$ with its corresponding workloads $\{wl_i | 1 \leq i \leq n\}$ and a set of m edge servers $ES = \{es_j | 1 \leq j \leq m\}$ with its corresponding upper service workloads $\{ub_j | 1 \leq j \leq m\}$. By incorporating the double auction scheme, each mobile device md_i claim bids b_{ij} to their interested edge servers es_j representing the most expected unit price for executing unit workload on es_j. The bidding values of each mobile device on the same edge server can be affected by several factors such as its proximity to the edge server, the quality of service on the edge server, etc. More importantly, we assume that each md_i can offload their workloads to different edge servers. The workloads can be executed parallel on different edge servers to improve efficiency. By contrast, each edge server es_j submits their ask price a_j representing the least expected unit price for executing unit workload on es_j. To be more specific, for all bids submitting to request computational resources on es_j, if the bidding value $b_{ij} < a_j$, then md_i is not allowed to join the auction on es_j. We assume that edge server es_j can serve multiple mobile devices at the same time but the total offloading workloads to es_j should not be bigger than its claimed capacity ub_j.

Our system acts as a trustworthy auctioneer and collects the information above provided by mobile devices (buyers) and edge servers (sellers). We not only provide the final offloading decision for these two entities but also decide the final pricing for them to clear the market. If the offloading decision has been finalized, we can get the final assigned workloads x_{ij} for md_i to es_j. Then A_j is denoted as the successful winning bids on es_j i.e. $A_j = \{b_{ij} | x_{ij} > 0\}$. In fact, in a double auction, there are two pricing aspects need to be defined for an assignment x_{ij}, the unit payment for md_i to es_j called p_j^i, and the unit revenue for executing unit workload on es_j called p_j. For those mobile devices with b_{ij}s in A_j, their final unit payment p_j^i should be the same.

To define the benefit gained on different entities and to motivate the concept of social welfare, the utility values of buyers and sellers are described as follows. The utilities on buyers can be explained as the amount of saved money. To be more specific, if a buyer md_i successfully win the bid for es_j with offloading workloads x_{ij}, then the amount of saved money on this allocation can be computed as $x_{ij} \cdot (b_{ij} - p_j^i)$. The total utility (total saving amount) of md_i can be aggregated as $U(md_i) = \sum_{j=1}^{j=m} x_{ij} \cdot (b_{ij} - p_j^i)$. In contrast, the utilities on sellers represent the additional revenues. Again, using the similar allocation case between md_i and es_j, the additional revenue for es_j on this allocation can be computed $x_{ij} \cdot (p_j - a_j)$. Then, the total utility (additional revenue) of es_j can be aggregated as $U(es_j) = \sum_{i=1}^{i=n} x_{ij} \cdot (p_j - a_j)$. Last, the social welfare value in the double auction is constituted by the summation of all buyers' and sellers' utilities, i.e. $\sum_{j=1}^{j=m} \sum_{i=1}^{i=n} x_{ij} \cdot (b_{ij} - p_j^i + p_j - a_j)$. After introducing the objective function of our problem, the integer programming model of our problem is formulated as follows.

TABLE I

Notation	Description
md_i	a mobile device
es_j	an edge server
b_{ij}	the bidding value on md_i to es_j
a_j	the ask price on es_j
wl_i	the workloads on md_i
ub_j	upper service workloads on es_j
x_{ij}	assigned workloads on md_i to es_j
p_j^i	finalized unit payment on md_i to es_j
p_j	finalized unit revenue on es_j
A_j	the successful winning bids on es_j

$$\max \sum_{j=1}^{j=m} \sum_{i=1}^{i=n} x_{ij} \cdot (b_{ij} - p_j^i + p_j - a_j)$$

$$\text{s.t.} \sum_i x_{ij} \leq ub_j \qquad \forall j \qquad (1)$$

$$\sum_j x_{ij} \leq wl_i \qquad \forall i \qquad (2)$$

$$p_j \geq a_j, p_j^i \leq b_{ij} \qquad \forall i, j \qquad (3)$$

$$x_{ij} \in \mathbb{N} \qquad \forall i, j \qquad (4)$$

Constraint (1) limits the upper workloads on edge servers. Constraint (2) implies a mobile device will not offload more than its workloads. Constraint (3) restrict the pricing to satisfy the *individual rationality*, a significant property in a double auction which will be explained later. Constraint (4) represents the assigned workloads should be a natural number. Table I summarizes the key notations used in this paper.

B. Desired properties in Double Auction

Some desired properties of a double auction need to be carefully considered in proposing an ideal double auction mechanism. These desired properties are stated as follows.

1) *Individual Rationality*:
Buyers and sellers should not lose from joining the auction. That is, $\forall i, j$, $p_j \geq a_j$, and $p_j^i \leq b_{ij}$.
2) *Strong Balanced Budget*:
After the trading between buyers and sellers, the auctioneer should not lose or gain money.
3) *Economic Efficiency*:
Several measurements might be selected for economic efficiency such as social welfare, the number of successful trades, or the revenue of edge servers, etc.
4) *Truthfulness*:
A double auction mechanism is truthful when an agent (buyer or seller) cannot improve its utility by reporting another untruthful bid.

In the following section, we demonstrate a double auction mechanism satisfying individual rationality, strong balanced budget, and economic efficiency. Even though all of the above properties cannot be achieved at the same time, as proved in [6], our work provides strong properties, including guaranteeing the optimality of social welfare, a popular objective in the task offloading problems in MEC and MCC (Mobile Cloud Computing) [7]–[11]. Social welfare is used because it best reflects the satisfaction level of the buyers and the sellers [12].

III. Our Proposed Double Auction Mechanism

In the design of our mechanism, we first try to satisfy the balance budget property. To this end, we make sure the transfers are done between buyers and sellers. Specifically, for each $x_{ij} > 0$, the prices on p^i_j and p_j can be directly set to be equal. For ease of our explanation, we utilize p_j to represent both the unit revenue of es_j and the unit payment for md_i who is assigned to offload its workload to es_j. By doing this, the auctioneer cannot gain or lose money and strong balance budget is satisfied. Moreover, the social welfare value can be re-formulated as $\sum_{j=1}^{j=m}\sum_{i=1}^{i=n} x_{ij} \cdot (b_{ij} - a_j)$, because each $x_{ij} > 0$, each pair of p^i_j and p_j can be summed up to be zero (i.e., $-p^i_j + p_j = 0$). In addition, the constraint (3) in the integer programming model can be modified as $a_j \leq p_j \leq b_{ij}$.

Individual Rationality is also a significant property in the double auction. It is not reasonable for buyers and sellers to lose from joining the auction. If the property is not satisfied, they might not be willing to join the auction due to dissatisfaction with their expected utilities. In this regard, before we start deciding the allocation, we first remove those bids with bidding values lower than its corresponding ask price on each edge server to avoid unnecessary operation on these bids. Then, constraint (3) can be satisfied (i.e., satisfying the *individual rationality*) if we let the final price of p_j locate within the range $[min(A_j), a_j]$ where $min(A_j)$ represents the minimum bidding value in A_j. In our mechanism, after we decide the winning bids, the finalized pricing will be directly set within the range $[min(A_j), a_j]$. In the following paragraph, we focus on winning bids determination, i.e. making the final offloading decision, for obtaining the optimal social welfare.

We convert the winning bids determination problem into a minimum cost flow problem [5], [13]. The minimum cost flow network is illustrated in Fig. 1. First, we create two sets of nodes for representing mobile devices and edge servers, i.e. $MD = \{md_i | 1 \leq i \leq n\}$ on the second layer in the network, and $ES = \{es_j | 1 \leq j \leq m\}$ on the third layer. Then, a source (or supply) node s and a destination (or demand) node d are used to produce the flow and to accept the flow respectively. We utilize f to represent the supply/demand status. If a node's $f > 0$, then it means that f flows are supplied from this node; If a node's $f < 0$, then $|f|$ flows are accepted to this node; those nodes with $f = 0$ are called transshipment nodes. On s node, we set $f = twl$ where $twl = \sum_{i=1}^{i=n} wl_i$, i.e. the total workloads. In contrast, we set $f = -twl$ to accept those flows. Then, the remaining nodes are all simply transshipment nodes. Generally speaking, we utilize the flows to represent the assigned workloads. The best flow status reaching the minimum cost can be interpreted to the best-assigned workloads reaching the optimal social welfare. In the following, we demonstrate how we set up the corresponding

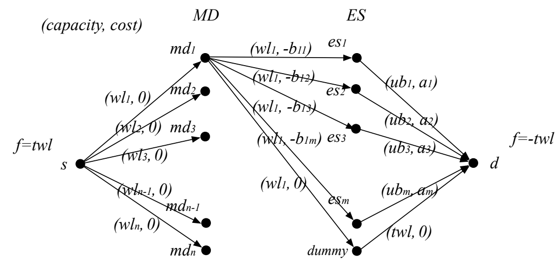


Fig. 1. Illustration of the minimum cost flow model.

capacity and cost on each link, the important parameters to clarify the constraints on assigned workloads and to bring the values to construct the social welfare.

In Fig. 1, we represent the corresponding capacity and cost on each link as (capacity, cost). The capacity on each link represents the upper amount of flows allowed to go through this link. If a flow goes through this link, then it will yield the corresponding cost set on this link. For the links between s and MD, we set up $(wl_i, 0)$ associated with each of these links. It means that each md_i can not be assigned more than its respective wl_i. For each link between MD and ES, $(wl_i, -b_{ij})$ is set up on the link from md_i to es_j. In Fig. 1, we simply illustrate the respective $(wl_1, -b_{1j})$ on links from md_1 to ES. Others can be easily set up in the same manner. Here, if md_i does not bid es_j or the bidding value b_{ij} is smaller than the ask value a_j, the link between them will not be created to avoid unnecessary operations. The capacity of wl_i is also used to restrict its upper assigned workloads. The links from md_i to ES represent that the workloads wl_i can be partitioned (offloaded) to different edge servers. The costs are set to the respective bidding values and change them to negative sign. This is mainly because the minimum cost flow is used to find the flow status with the minimum cost. We will formally explain that the finalized flow status reaching the minimum cost can be regarded as the finalized offloading result reaching the optimal social welfare in Section IV. For each link from ES to d, (ub_j, a_j) is set respectively. Obviously, ub_j is set to constrain the upper workloads on edge server es_j, and it will gain a_j cost if any single flow goes through this link. One can notice that we create a dummy node on the third layer. The dummy node is used to accommodate the residual flows $twl - \sum_{j=1}^{j=m} ub_j$ in the case that $twl > \sum_{j=1}^{j=m} ub_j$. The costs of those links linked to this dummy node are set to zero, and the capacities should be set accordingly, as illustrated in Fig. 1. By building this minimum cost flow network model, we can successfully deal with the winning bids determination problem. Specifically, we adopt a state-of-the-art algorithm [5] to deice the finalized flow obtaining the minimum cost. The finalized flows on each link between md_i to es_j is then interpreted as the finalized offloading workloads x_{ij}.

Motivating Example of a Solution Across Distributed Servers: We use a toy example to show that the workloads of a user will be indeed distributed into different edge servers in some cases to obtain the optimal social welfare. In the toy example, $MD = \{md_1, md_2\}$ and the respective workloads are $wl_1 = 2, wl_2 = 1$. $ES = \{es_1, es_2\}$ and the respective upper service workloads are $ub_1 = 1, ub_2 = 2$. The ask prices are set as $a_1 = 3, a_2 = 4$. The bids are $b_{11} = 5, b_{12} = 8, b_{21} = 6, b_{22} = 6$. By using the minimum cost flow model, the result of the winning bids will be: $x_{11} = 1$, $x_{12} = 1$, $x_{22} = 1$. The workloads of md_1 will be distributed into different edge servers to reach the optimal social welfare.

IV. Theoretical Analysis

We formally prove our double auction mechanism achieving individual rationality, strong balance budget, economic efficiency on social welfare in this section.

Theorem 1. *Our mechanism achieves individual rationality.*

Proof. For buyers with no winning bids and sellers with no assigned workloads, there is no trade among them. Hence they do not have any chance to lose. For a buyer md_i with a winning bid b_{ij}, the finalized unit payment p_j is set to a_j. Because $b_{ij} \geq p_j$, md_i will not lose from joining the auction. Similarly, for a seller es_j with some assigned workloads, the finalized unit revenue of p_j is set to a_j. With this setting $p_j = a_j$, es_j will not lose from joining the auction. □

Theorem 2. *Our mechanism achieves strong balanced budget.*

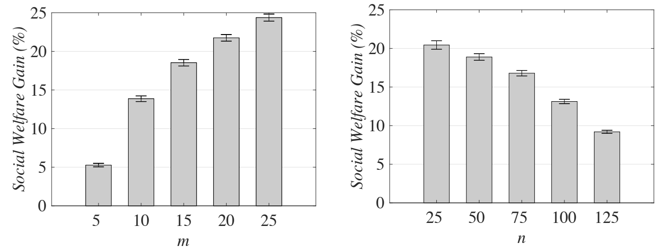
Proof. After the winning bids determination stage, a winning buyer md_i is assigned to offload its workloads x_{ij}s to different edge servers with the finalized unit payments. If es_j is one of its offloading servers, the finalized payment for md_i on edge server es_j should be $x_{ij} \cdot p_j$. The payment will be directly transferred to es_j, i.e. the same amount $x_{ij} \cdot p_j$ will be received on es_j. Hence, an auctioneer does not lose or gain money. □

Theorem 3. *Our mechanism can always find the best winning bids obtaining optimal social welfare in polynomial time.*

Proof. We reduce our problem to a minimum cost flow problem which can be solved by a state-of-the-art algorithm in polynomial time [5]. By utilizing the minimum cost flow model to reach the optimal social welfare, we change the bidding values to negative sign. In such a case, the objective function is modified from maximizing $\sum_{j=1}^{j=m} \sum_{i=1}^{i=n} x_{ij} \cdot (b_{ij} - a_j)$ to minimizing $\sum_{j=1}^{j=m} \sum_{i=1}^{i=n} x_{ij} \cdot (-b_{ij} + a_j)$. The algorithm decides the finalized x_{ij} with the guarantee of minimum cost and it can be inferred that it can reach the optimal social welfare at the same time due to the negative sign. □

V. Performance evaluation

The performance evaluation of our proposed mechanism is focused on the economic efficiency, i.e., the social welfare, and the computational efficiency, i.e., the running time for the winning bids determination. The simulations are implemented in Python and conducted on an Intel Core i7 CPU machine with 8GB memory. To solve the minimum cost flow problem, we utilized the Google OR-tools [14] implementing the state-of-the-art cost-scaling push-relabel algorithm [5]. As discussed in [15], the solver of Google OR-tools could solve the problem more efficiently compared to other tools. We conducted several simulations under different parameter settings. All results also include 95% confidence intervals. The default value of n is set to 50, and m is set to 15. In the following, the random



(a) Effect of different m (b) Effect of different n

Fig. 2. Social Welfare Gain

selection of values basically follows the uniform distribution. Each md_i will randomly create its bids b_{ij} within the range of [0,14] to m edge servers and randomly select its workloads wl_i within the range of [10, 30]. If $b_{ij} = 0$, it means that md_i is not under the service range provided by es_j or is not interested in joining es_j. Each es_j will randomly generate its ask value a_j within the range of [3,10] and randomly choose the serving upper bound on workloads ub_j within the range of [100, 300]. If $b_{ij} < a_j$, then md_i loses the chance to compete for offloading its workloads to es_j. As we stated in section III, the link between them will not be generated in the network model to avoid unnecessary operations.

1) Economic Efficiency (Social Welfare): Our mechanism can always reach the optimal social welfare proved in Theorem 3. Because we are the first to use double auction and optimize social welfare, there is no suitable mechanism to compare with our mechanism. However, we still want to reveal our remarkable results on social welfare against some other approach. In this regard, we selected the DPDA method proposed in [3] because the scenario in [3] is the closest to ours. Specifically, existing works using double auction in MEC or MCC assume that each edge server can only serve one mobile device [16], or each edge server can serve multiple devices [3]. Undoubtedly, the latter one is closer to our proposing scenario. However, each md_i can only offload one task to one edge server in [3]. To let their method can still serve for the workload case, we separate the workloads to several mobile devices for fitting their approach in which each mobile device can only offload one task (one workload). Specifically, if $wl_1 = 10$, we will create ten virtual mobile devices. By doing this, their method can still be utilized to solve our proposed scenario. However, their objective function is not social welfare but rather the number of successful trade, which is not a popular approach in MEC. Undoubtedly, their DPDA method based on a greedy approach can not guarantee to reach the optimality. We then introduce a social welfare gain to present the percentage gain between the obtained social welfare of DPDA and our optimal social welfare. Then, we investigate the effects of different m and n on the social welfare gain in Fig. 2.

We can observe a consistent improvement in social welfare gain in Fig. 2(a). This is because when m is increased, the combinatorial nature becomes more significant. It is much difficult for DPDA to obtain good social welfare because the greedy approach lacks of a comprehensive view, just greedily assign according to the bidding values. It will lose some matchings among them due to the upper bound constraints on

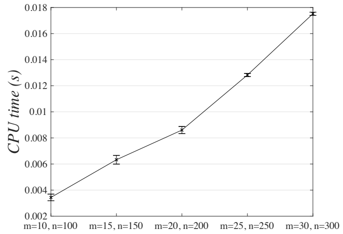


Fig. 3. Scalability of our mechanism

edge servers. In Fig. 2(b), we fix $m = 15$ while varying n from 25 to 125. The increasing of n also increases the combinations, but it is not as significant as increasing on m. The bigger number of n implies that a bigger portion of the capacities on edge servers is occupied. It means that it will make the obtained social welfare obtained by the greedy approach close to the optimal value. The result in Fig. 2(b) shows that the trend of social welfare gain is decreased. It can be explained as the second argument has a more significant impact than the combinatorial issue. Nevertheless, as proved in Theorem 3, our mechanism can always obtain the optimal social welfare.

2) Computational Efficiency: For computational efficiency, we focus on evaluating computational performance by showing the scalability. We increase m and n together in Fig. 3. In fact, when we increase m or n separately (while fixing the other), the running time also monotonically increases, which is similar to Fig. 3. Increasing m and n simultaneously is a reasonable approach for evaluating the scalability. In a practical scenario, if one of them increases, it should drive the number of the other one. Our mechanism provides great computational performance among different scales of the system.

VI. Related Work

Auction schemes are widely utilized for the offloading problems in MEC [3], [8], [11] and other relevant areas: mobile cloud computing (MCC) [16], and heterogeneous resources allocation involving both fog cloud, edge server, etc. [9]. We focus on the literature using the double auction for MEC. In [3], Sun et al. consider each edge server can serve multiple mobile devices with its claimed upper bound. However, each mobile device can only offload its task to one edge server. They proposed two double auction mechanisms, a breakeven-based approach, and a more efficient dynamic pricing based approach. However, the objective function they selected is to maximize the number of successfully assigned pairs, which is not a common objective selected for economic efficiency. In our problem, we choose the most popular objective, i.e., social welfare [7]–[11]. Even though Yue et al. [8] select social welfare as the metric, they do not reach the optimality. They mainly focus on satisfying the other three properties. Also, they model the allocation problem from the task perspective, where each mobile device has different types of tasks to be offloaded. In our work, we model the problem from the workload perspective, which is more aligned with current edge service providers' pricing models [2].

VII. Conclusion and Future Work

We propose a double auction mechanism for the task offloading problem in MEC. To better model the real-world practices, we not only allow many-to-many allocation but also adopt a proportional pricing scheme to support the current pricing scheme on Amazon Lambda@Edge [2]. We successfully convert the winning bids determination problem into a minimum cost flow problem. A state-of-the-art algorithm [5] utilized for solving minimum cost flow decides the final task offloading result with optimal social welfare. Based on the theoretical analysis and the simulation results, our mechanism can efficiently obtain optimal social welfare while satisfying individual rationality and strong balance budget. Future research directions to improve our double-auction scheme include: the investigation of the trade-off between truthfulness and social welfare (for example, guaranteeing truthfulness while achieving near-optimal social welfare), an online algorithm to serve a more dynamic scenario in MEC, the systems implementation of the scheme (including the efficient delivery mechanisms of the workloads across distributed servers).

Acknowledgment

This research was supported in part by Colorado State Bill 18-086.

References

[1] N. Abbasa, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, Feb 2018.

[2] Amazon Web Services. (2019) Lambda@edge. [Online]. Available: https://aws.amazon.com/lambda/edge/

[3] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4692–4701, Oct 2018.

[4] J. Zhang, D. Wen, and S. Zeng, "A discounted trade reduction mechanism for dynamic ridesharing pricing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1586–1595, June 2016.

[5] A. V. Goldberg, "An efficient implementation of a scaling minimum-cost flow algorithm," *Journal of Algorithms*, vol. 22, no. 1, pp. 1 – 29, 1997.

[6] R. B. Myerson and M. A. Satterthwaite, "Efficient mechanisms for bilateral trading," *Journal of Economic Theory*, vol. 29, no. 2, pp. 265 – 281, 1983.

[7] Z. Zhao, F. Chen, T. H. Chan, and C. Wu, "Double auction for resource allocation in cloud computing," in *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER)*, 2017, pp. 273–280.

[8] Y. Yue, W. Sun, and J. Liu, "Multi-task cross-server double auction for resource allocation in mobile edge computing," in *Proceedings of the 53rd IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[9] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju, "Decloud: Truthful decentralized double auction for edge clouds," in *Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 07 2019.

[10] L. Y. Chu and Z. M. Shen, "Truthful double auction mechanisms," *Operations Research*, vol. 56, no. 1, pp. 102–120, 2008.

[11] H. Zhang, F. Guo, H. Ji, and C. Zhu, "Combinational auction-based service provider selection in mobile edge computing networks," *IEEE Access*, vol. 5, pp. 13 455–13 464, 2017.

[12] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, 2007.

[13] J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Operations Research*, vol. 41, no. 2, pp. 338–350, 1993.

[14] Inc. Google. (2018) Google optimization tools (or-tools). [Online]. Available: https://github.com/google/or-tools

[15] X. Liu, Y. Liu, and Z. Yang, "Paper review assignment problem under the minimum cost flow network model," in *Proceedings of the Twenty-Eighth RAMP Symposium*, October 2016.

[16] A. L. Jin, W. Song, and W. Zhuang, "Auction-based resource allocation for sharing cloudlets in mobile cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 45–57, Jan 2018.