A Double Auction Mechanism for Coded Distributed Computing in Smart Vehicles

Jer Shyuan Ng[1,2], Wei Yang Bryan Lim[1,2], Zehui Xiong[3], Sahil Garg[4], Yang Zhang[5], Dusit Niyato[6] and Cyril Leung[7,8]

[1]*Alibaba Group, Singapore*

[2]*Alibaba-NTU Joint Research Institute (JRI), Singapore*

[3]*Pillar of Information Systems Technology and Design, Singapore University of Technology Design, Singapore*

[4]*École de Technologie Supérieure, Montreal, Canada*

[5]*Nanjing University of Aeronautics and Astronautics, China*

[6]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*

[7]*LILY Research Center, Nanyang Technological University, Singapore*

[8]*Department of Electrical and Computer Engineering, The University of British Columbia, Canada*

Keywords: Coded Distributed Computing, Double Auction, Straggler Effects, Resource Allocation.

Abstract: The development of smart vehicles and rich cloud services have led to the emergence of vehicular edge computing. To perform the distributed computation tasks efficiently, Coded Distributed Computing (CDC) was proposed to reduce communication costs and mitigate the straggler effects through the use of coding techniques. In this paper, we propose a double auction mechanism to allocate the resources of the edge servers to the vehicles in order to complete the CDC tasks. Specifically, the vehicles use the PolyDot codes to manage the tradeoff between communication costs and recovery threshold. Given the requirements of various vehicles, the double auction mechanism matches the edge servers with the required resources to the vehicles. Besides, the double auction mechanism also determines the prices that the vehicles need to pay for the resources of the edge servers. The double auction mechanism satisfies the properties of individual rationality, incentive compatibility and budget-balance.

1 INTRODUCTION

With the advent of vehicular applications, there has been a significant transformation in the way that data are managed, where we observe a paradigm shift from the traditional centralized cloud computing toward flexible distributed vehicular edge computing. Instead of processing the data at the central cloud server, the computing and storage services are offloaded to the vehicles, extending the cloud computing services to the edge of the networks. Instead of performing the large-scale computations individually, the vehicles can leverage on the resources of the edge servers, e.g., roadside units (RSUs) and base stations, to complete their computation tasks efficiently over the distributed vehicular edge computing networks (Xu et al., 2019).

Coded Distributed Computing (CDC) has been proposed to perform distributed computation tasks efficiently. Coding techniques can be used to achieve an optimal tradeoff between computation and communication loads, thus reducing the communication costs required for the distributed computation tasks. Besides, the computation strategies are designed to divide and allocate the datasets to the edge servers so that the vehicles are able to reconstruct the final result with the computed results from only a subset of edge servers. By reducing the number of edge servers required to return their results, the computation latency can be minimized.

However, the edge servers are not motivated to facilitate the CDC tasks of the vehicles. As such, incentive mechanisms are essential to encourage the edge servers to complete the CDC tasks by offering them monetary rewards in exchange for their resources. In order to ensure efficient completion of the CDC tasks, the incentive mechanisms need to allocate the CDC tasks to the edge servers and determine the payment prices to the edge servers. Since there are multiple vehicles competing for the resources of the edge servers, we propose a double auction mechanism to model the competition between different vehicles and edge servers as well as to maximize the utility of the auctioneer.

Ng, J., Lim, W., Xiong, Z., Garg, S., Zhang, Y., Niyato, D. and Leung, C.
A Double Auction Mechanism for Coded Distributed Computing in Smart Vehicles.
DOI: 10.5220/0010615801070114
In *Proceedings of the 18th International Conference on Wireless Networks and Mobile Systems (WINSYS 2021)*, pages 107-114
ISBN: 978-989-758-529-6

In this paper, we consider that there are multiple vehicles that aim to perform machine learning tasks using the data they collected. Due to the lack of computation resources, the vehicles leverage on the computation and communication resources of the edge servers. In order to perform the distributed computation tasks efficiently, the vehicles use PolyDot codes (Dutta et al., 2020) to divide and allocate the dataset to the edge servers. By using PolyDot codes, the vehicles are able to determine the number of edge servers that they require to complete their CDC tasks. Specifically, for time-sensitive applications, the vehicles aim to minimize the straggler effects and hence need fewer edge servers to compute the CDC tasks. However, since each of the edge servers is allocated a larger size of dataset for computation, larger communication costs are needed to transmit the computed results and thus, the vehicles need to pay higher prices to the edge servers. Hence, there is a tradeoff between the recovery threshold, i.e., the number of edge servers that need to return their computed results to the vehicles in order to recover the final result, and the communication costs. Each vehicle requires different number of edge servers, depending on their urgency to complete their CDC tasks. Given the different objectives of the vehicles and the heterogeneity of the edge servers, the double auction mechanism matches the edge servers to the vehicles as well as determines the payment and selling prices. The double auction mechanism is managed by an auctioneer, which is represented by a third-party platform, that aims to maximize its total profit. Thus, the payment prices of the buyers for the resources are larger than the selling prices of the sellers. In other words, the profit of the auctioneer for each match of winning buyer and seller is the difference between the payment and selling price of the winning buyer and seller respectively.

2 SYSTEM MODEL AND PROBLEM FORMULATION

The system model comprises a set $\mathcal{M} = \{1,\dots,m,\dots,M\}$ of M vehicles and a set $\mathcal{N} = \{1,\dots,n,\dots,N\}$ of N edge servers in a distributed vehicular edge computing network. Equipped with more sophisticated sensors, the vehicles collect data from their surroundings, which offer meaningful insights, e.g., by using machine learning techniques (Radu et al., 2020), (Fusco et al., 2015). Given the massive amount of constantly changing data, AI models have shown their effectiveness in obtaining valuable information from every piece of vehicular data generated, ranging from the understanding behaviour pattern of consumers of a particular in-vehicle application to improving the productivity of businesses to monitoring large-scale phenomena such as tracking of road conditions. However, the vehicles may not have sufficient resources, e.g., CPU power, to handle the growing datasets individually. Instead, the vehicles can leverage on the resources of the edge servers to facilitate their computation tasks, e.g., training of AI models.

2.1 Coded Distributed Computing

We consider that each vehicle m aims to compute the matrix multiplication of two $Q_m \times Q_m$ square input matrices $\mathbf{A}^m$ and $\mathbf{B}^m$, i.e., $\mathbf{C}^m = \mathbf{A}^m\mathbf{B}^m$. In order to mitigate the straggler effects in performing the distributed computation tasks, the vehicles can adopt coding techniques to divide the datasets and allocate the subsets of data to the edge servers. The objective of various coding techniques is to minimize the recovery threshold, i.e., the number of edge servers that need to return their computed results to the vehicles in order to recover the final result. However, a smaller recovery threshold means that there are fewer edge servers that are working on the computation task. Hence, each edge server is allocated a larger dataset to compute and has greater communication costs in terms of number of symbols communicated to the vehicles.

To manage the tradeoff between the recovery threshold and the communication costs of the edge servers, PolyDot codes (Dutta et al., 2020) are proposed where Polynomial codes (Yu et al., 2017) and MatDot codes (Dutta et al., 2020) are special instances of this coding framework by considering the extreme cases: to either minimize communication costs or recovery threshold. In particular, Polynomial codes achieve minimum recovery threshold at the expense of higher communication costs whereas MatDot codes minimize communication costs but require larger recovery threshold.

Unlike Polynomial codes and MatDot codes that split the input matrices either horizontally or vertically, each vehicle m uses PolyDot codes to split the input matrices $\mathbf{A}^m$ and $\mathbf{B}^m$ both horizontally and vertically such that:

$$\mathbf{A}^m = \begin{bmatrix} \mathbf{A}^m_{0,0} & \cdots & \mathbf{A}^m_{0,s_m-1} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^m_{t_m-1,0} & \cdots & \mathbf{A}^m_{t_m-1,s_m-1} \end{bmatrix},$$
$$\mathbf{B}^m = \begin{bmatrix} \mathbf{B}^m_{0,0} & \cdots & \mathbf{B}^m_{0,t_m-1} \\ \vdots & \ddots & \vdots \\ \mathbf{B}^m_{s_m-1,0} & \cdots & \mathbf{B}^m_{s_m-1,t_m-1} \end{bmatrix}, \tag{1}$$

where $\mathbf{A}^m_{j,i}$ and $\mathbf{B}^m_{i,j}$ are $Q_m/t_m \times Q_m/s_m$ and $Q_m/s_m \times Q_m/t_m$ submatrices respectively for $i = \{0,1,\ldots,s_m-1\}$ and $j = \{0,1,\ldots,t_m-1\}$. The values of parameters s_m and t_m are chosen such that $s_m t_m = \omega_m$, where ω_m is the number of submatrices of vehicle m.

For PolyDot codes, the encoding polynomials $p^m_{\mathbf{A}}(x,y)$ and $p^m_{\mathbf{B}}(y,z)$ for each vehicle m are as follows:

$$p^m_{\mathbf{A}}(x,y) = \sum_{i=0}^{t_m-1}\sum_{j=0}^{s_m-1} \mathbf{A}^m_{i,j} x^i y^j,$$
$$p^m_{\mathbf{B}}(y,z) = \sum_{k=0}^{s_m-1}\sum_{l=0}^{t_m-1} \mathbf{B}^m_{k,l} y^{s_m-1-k} z^l, \tag{2}$$

where variables x, y and z are assigned to the encoding polynomials where each of the encoding polynomials $p^m_{\mathbf{A}}(x,y)$ and $p^m_{\mathbf{B}}(y,z)$ is assigned two variables. The encoding polynomials with three variables are transformed to polynomials with a single variable by choosing $y = x^{t_m}$ and $z = x^{t_m}(2s_m-1)$.

There are three major steps in completing the CDC tasks:

1. **Encoding:** The vehicles divide and encode the submatrices. Then, the vehicles distribute the encoding polynomials to the edge servers. Each edge server n receives $p^m_{\mathbf{A}}(x_n,y_n)$ and $p^m_{\mathbf{B}}(y_n,z_n)$ where $y_n = x_n^{t_m}$ and $z_n = x_n^{t_m(2s_m-1)}$.
2. **Computation:** Each edge server n computes the matrix multiplication of the encoding polynomials and produces the result $\mathbf{C}^m(x_n,y_n,z_n) = p^m_{\mathbf{A}}(x_n,y_n)p^m_{\mathbf{B}}(y_n,z_n)$. Upon completion, the edge servers transmit the result back to the vehicles.
3. **Decoding:** Each vehicle m is able to reconstruct the final result $\mathbf{C}^m = \mathbf{A}^m\mathbf{B}^m$ upon receiving the computed results from any $t_m^2(2s_m-1)$ edge servers. In particular, the vehicle uses the computed results from the edge servers to recover the coefficient of $x^{i-1}y^{s_m-1}z^{l-1}$ in $\mathbf{C}^m(x,y,z)$. By using the transformed single-variable encoding polynomials, the vehicles compute the coefficient of $x^{i-1+t_m(s_m-1)+t(2s_m-1)(l-1)}$ in $\mathbf{C}^m(x,x^{t_m},x^{t_m}(2s_m-1))$.

By using PolyDot codes, the recovery threshold, k_m, that can be achieved by vehicle m is defined as follows:
$$k_m = t_m^2(2s_m-1). \tag{3}$$

To complete the CDC subtasks of vehicle m, the number of symbols that each edge server is required to transmit, which is denoted as μ_m, is represented as follows:

$$\mu_m = \frac{Q_m^2}{t_m^2}. \tag{4}$$

Hence, in order to complete the CDC task of vehicle m, the total number of symbols communicated by the edge servers to vehicle m is $\frac{Q_m^2}{t_m^2} \times t_m^2(2s_m-1) = (2s_m-1)Q_m^2$.

The number of computations that each edge server requires to complete the CDC subtasks of vehicle m, which is denoted as α_m, is defined as follows:

$$\alpha_m = \frac{Q_m^3}{s_m t_m^2}. \tag{5}$$

The total number of computations that are performed by the edge servers that facilitate in the CDC task of vehicle m is $\frac{Q_m^3}{s_m t_m^2} \times t_m^2(2s_m-1) = \frac{Q_m^3}{s_m}(2s_m-1)$.

In our system model, we consider the matrix multiplication of two square matrices. Matrix multiplication is an important basic mathematical operation for many useful AI algorithms. It may involve the multiplication of non-square matrices or more than two matrices. Our system model can be easily extended to compute the matrix multiplication of non-square matrices and more than two matrices by using PolyDot codes.

2.2 Double Auction Model

Since there are multiple vehicles and edge servers in the network, a double auction mechanism is adopted. The objectives of the double auction mechanism are (i) to match the edge servers to the vehicles, and (ii) to determine the payment prices of vehicles and the selling prices of edge servers.

In this double auction mechanism, there are several entities, where their roles are described as follows:

1. **Vehicles (Buyers):** The vehicles are buyers that pay the edge servers for their resources in facilitating the CDC subtasks. In order to participate in the auction, the vehicles submit their buy-bids to the auctioneer. The buy-bid of vehicle m, which is denoted by b_m, represents the maximum price that vehicle m is willing to pay for each edge server to

complete its CDC subtask. The vehicles are able to decide on the recovery threshold and the communication costs required to complete the CDC tasks by using the PolyDot codes. Specifically, for vehicles that require their CDC tasks to be completed within a shorter period of time, they need a smaller recovery threshold, which in turn results in higher communication costs required by each edge server. As such, the vehicles need to pay more for the edge servers to complete the CDC subtasks. Conversely, for vehicles that need to complete CDC tasks without time constraint, they aim to reduce communication costs of the edge servers in order to reduce the payment prices to the edge servers. The buy-bid of vehicle m is expressed as follows:

$$b_m = \lambda_m^1 \mu_m + \lambda_m^2 \alpha_m, \tag{6}$$

where λ_m^1 and λ_m^2 are the unit prices that vehicle m is willing to pay for the communication and computation resources respectively.

2. **Edge Servers (Sellers):** The edge servers are sellers that provide their computation and communication resources to facilitate the CDC subtasks of the vehicles. In return, they receive monetary rewards for the provision of their resources. By participating in the double auction mechanism, the sellers submit their sell-bids to the auctioneer. The amounts of computation and communication resources of each edge server n are represented by r_n^{cp} and r_n^{cm} respectively. The unit costs of computation and communication resources are represented by c_n^{cp} and c_n^{cm} respectively. Based on the values of these parameters, the sell-bid of seller n, which is the price that the seller n wants to receive for providing its resources to complete the CDC subtask, can be computed as follows:

$$q_n = r_n^{cm} c_n^{cm} + r_n^{cp} c_n^{cp}. \tag{7}$$

3. **Auctioneer:** Typically, the service provider platform plays the role of the auctioneer as a trusted third-party in order to ensure fairness and transparency of the transaction between the buyers and sellers. The auctioneer collects the buy-bids and sell-bids from the buyers and sellers respectively. Then, it allocates the edge servers to the vehicles. Besides, the auctioneer determines the payment price of the vehicles for the resources of the edge servers. In order to prevent malicious buyers and sellers from threatening the security of the network, the auctioneer can also implement an authentication system (Lin et al., 2012) (Liu et al., 2014) to ensure the trustworthiness of the buyers and sellers. The auctioneer can be regarded as the unified API that aggregates the resources of the edge servers and provides the vehicles with their required resources for a price, a fraction of which is paid to the corresponding edge servers that facilitate the CDC tasks.

The utility of buyer m, which is denoted as u_m, is expressed as follows:

$$u_m = \begin{cases} k_m(b_m - p_m), & \text{if } b_m \in \mathcal{W}_b, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where p_m payment price of buyer m. The set $\mathcal{W}_b$ consists of the buy-bids of the winning buyers. If buyer m is not matched with any seller, the utility of buyer m is zero.

The utility of seller n is denoted as follows:

$$u_n = \begin{cases} \gamma_n - q_n, & \text{if } q_n \in \mathcal{W}_q, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where γ_n is the price paid by the auctioneer to seller n for its resources. The set $\mathcal{W}_q$ consists of the winning sell-bids of successfully matched sellers. If seller n is not successfully matched to any buyer, its utility is zero.

The double auction mechanism is managed by the third-party auctioneer. The utility of the auctioneer, u_a, is the difference between the payment received by all successfully matched buyers and the payment paid to all successfully matched sellers, which is expressed as follows:

$$u_a = \sum_{b_m \in \mathcal{W}_b} k_m p_m - \sum_{q_n \in \mathcal{W}_q} \gamma_n. \tag{10}$$

The design of the double auction mechanism has several properties:

- Individual Rationality: Each buyer and seller achieves individual rationality in this double auction mechanism. In other words, each of the buyers and sellers receives non-negative utility by participating in the auction. In particular, each buyer m pays a price that is lower than or equal to its buy-bid, i.e., $b_m - p_m \geq 0, \forall m \in \mathcal{M}$, and each seller n receives a payment that is higher than or equal to its sell-bid, i.e., $q_n - \gamma_n \leq 0, \forall n \in \mathcal{N}$.
- Incentive Compatibility: There is no incentive for the buyers and sellers to submit buy-bids and sell-bids respectively other than their true valuations. In other words, the buyers and sellers submit their bids truthfully.
- Budget-balance: As a third-party platform that determines the allocation of edge servers to the vehicles and the corresponding payment and selling

prices, the auctioneer needs to gain a non-negative utility. Specifically, the total payment price that is received from all winning bidders is higher than or equal to the total price paid to all winning sellers, i.e., $u_a \geq 0$.

The objectives of this double auction mechanism are to match the buyers and sellers as well as to determine the payment and selling prices of the buyers and sellers respectively in order to complete the CDC tasks. As a platform that manages the transactions between buyers and sellers, the aim of the auctioneer, which is to maximize its total profit and thus utility while ensuring that the double auction mechanism is individually-rational and incentive-compatible, can be expressed as follows:

$$\begin{aligned} \max \quad & u_a \\ \text{s.t.} \quad & b_m \geq p_m, && \forall m \in \mathcal{M}, \\ & q_n \leq \gamma_n, && \forall n \in \mathcal{N}, \\ & b_m - p_m \geq b'_m - p'_m, && \forall m \in \mathcal{M}, \\ & \gamma_n - q_n \geq \gamma'_n - q'_n, && \forall n \in \mathcal{N}, \end{aligned} \tag{11}$$

where b'_m and q'_n are the untruthful buy-bid of buyer m and sell-bid of seller n respectively. The terms p'_m and γ'_n are the payment price and selling price, given the untruthful buy-bid b'_m and q'_n respectively.

3 DOUBLE AUCTION MECHANISM

In this section, we discuss the design of the double auction mechanism. There are two important stages in the double auction mechanism: matching of the buyers and sellers and the determination of the payment prices of buyers and the selling prices of sellers.

3.1 Matching between Buyers and Sellers

Given the buy-bids from the buyers, the auctioneer sorts the buy-bids in the descending order, which is represented by the set $\mathcal{B} = \{b_m^{(1)}, \dots, b_m^{(m)}, \dots, b_m^{(M)}\}$, where:

$$b_m^{(1)} \geq b_m^{(2)} \geq \cdots \geq b_m^{(M)}, \tag{12}$$

and $b_m^{(1)}$ and $b_m^{(M)}$ are the largest and smallest buy-bids which are offered by buyer m respectively.

The sell-bids from the sellers are sorted in the ascending order in set $\mathcal{S} = \{q_n^{(1)}, \dots, q_n^{(n)}, \dots, q_n^{(N)}\}$ where:

$$q_n^{(1)} \leq q_n^{(2)} \leq \cdots \leq q_n^{(N)}, \tag{13}$$

and $q_n^{(1)}$ and $q_n^{(N)}$ are the smallest and largest sell-bids that are submitted by seller n respectively .

Given the sorted buy-bids and sell-bids, the auctioneer matches the vehicles and edge servers that satisfy the following inequalities:

$$b_m \geq q_n, \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{14}$$

$$\frac{r_n^{cm}}{g_n} \geq \mu_m, \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{15}$$

$$\frac{r_n^{cp}}{h_n} \geq \alpha_m, \quad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{16}$$

where g_n is the amount of communication resources needed to transmit each symbol and h_n is the amount of computation resources needed for each computation. The inequality (14) requires the buy-bid of buyer m to be larger than or equal to the sell-bid of the seller n for a successful match in order to satisfy the property of individual rationality of the auction mechanism. Inequalities (15) and (16) ensure that the seller n has sufficient communication and computation resources to perform the CDC subtasks of buyer m respectively. If there are multiple sellers that satisfy that conditions for a particular buyer m, the seller with the smallest sell-bid wins.

The match results between the buyers and sellers can be represented by a matching matrix, which is expressed as follows:

$$\beta = \begin{bmatrix} \beta_1^1 & \beta_1^2 & \cdots & \beta_1^N \\ \beta_2^1 & \beta_2^2 & \cdots & \beta_2^N \\ \vdots & \vdots & \ddots & \vdots \\ \beta_M^1 & \beta_M^2 & \cdots & \beta_M^N \end{bmatrix}, \tag{17}$$

where $\beta_m^n = 1$ if seller n is matched to buyer m whereas $\beta_m^n = 0$ if the match between buyer m and seller n is not successful.

3.2 Determination of Payment and Selling Prices

In order to determine the payment prices of the winning buyers to the auctioneer as well as the selling prices of the winning sellers, we present a pricing scheme which is similar to the second-price auction. In particular, if buyer m is matched with seller n, $\forall n \in \mathcal{N}$, the payment price of winning buyer m equals the next largest buy-bid, given its buy-bid. The selling price of winning seller n equals the next smallest sell-bid in the feasible set of sellers, i.e., the sellers with sell-bids that fulfil inequalities (14), (15) and (16) and thus can possibly matched to buyer m, given the seller's sell-bid.

The difference between the payment price of winning buyer m and selling price of winning seller n is equal to the profit of the auctioneer.

4 SIMULATION RESULTS

We consider a distributed vehicular edge computing network that consists of 6 vehicles and 50 edge servers. The values of system simulation parameters are summarized in Table 1. We set $Q_m = 10$ and $w_m = 4$, $\forall m \in \mathcal{M}$, where all vehicles have the input matrices of the same size and split the input matrices into 4 submatrices. Given the requirements of the vehicles, they split their input matrices horizontally and vertically and encode their submatrices using PolyDot codes. The splits of input matrices are determined by variables s_m and t_m as shown in Equation (1).

Table 1: System Simulation Parameter Values.

Parameter	**Values**
Unit cost of communication resources, c_n^{cm}	$[0.1, 1]$
Unit cost of computation resources, c_n^{cp}	$[0.1, 1]$
Total amount of communication resources, r_n^{cm}	1-50 Gbps
Total amount of computation resources, r_n^{cp}	10-200MHz
Amount of communication resources required for each transmission, g_n	100-500Mbps
Amount of computation resources required for each computation, h_n	100-500kHz
Unit price of communication resources, λ_m^1	[0.1, 1]
Unit price of computation resources, λ_m^2	[0.1, 1]

Table 2: Simulation Parameter Values of Buyers.

Buyer ID	s_m	t_m	k_m	λ_m^1	λ_m^2
Buyer 1	1	4	16	0.6	0.5
Buyer 2	1	4	16	0.5	0.4
Buyer 3	2	2	12	0.3	0.8
Buyer 4	2	2	12	0.4	0.4
Buyer 5	4	1	7	0.4	0.5
Buyer 6	4	1	7	0.4	0.3

The values of the simulation parameters of the 6 buyers are shown in Table 2. The buyers with larger t_m and smaller s_m have larger recovery threshold. Given the number of symbol transmissions and computations required for the CDC tasks as well as the unit

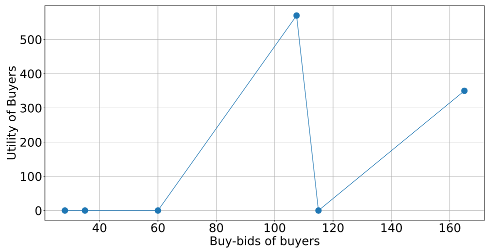

Figure 1: Utility of Buyers.

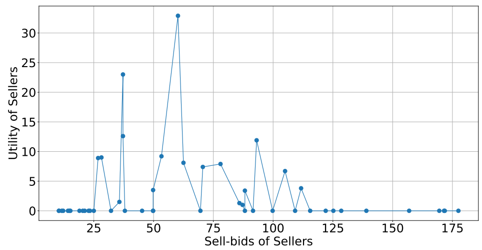

Figure 2: Utility of Sellers.

prices that the buyers are willing to pay for the resources, the buy-bids of the buyers can be computed.

Figure 1 shows the utilities of the buyers given the buy-bids of the buyers. We observe that for buyers that submit buy-bids of lower than 60, their utilities are zero. This implies that by bidding a value of smaller than 60, the buyers are unable to complete their CDC tasks as they do not have sufficient edge servers to facilitate their CDC tasks. There are several reasons for the failure in their bids for edge servers. Firstly, the buyers are not matched to sellers that have larger sell-bids than the buyers' buy-bids to ensure individual rationality of the double auction mechanism. Thus, only a small number of sellers that can be matched to buyers with smaller bids. Secondly, since each edge server can only facilitate the CDC task of one of the vehicles, the buyers with small bids lose out to those with large bids in the auction, reducing further the number of sellers that can possibly match to the buyers with small bids. Besides, we observe that the buyer with a buy-bid of 115 receives a utility of zero but the buyer with a smaller buy-bid of 107.5 receives a utility of 570, resulting in a discontinuity in the curve in Fig. 1. The vehicles require a number of edge servers, which is defined by the recovery threshold, to complete their CDC tasks. However, there may not be enough edge servers that have sufficient communication and computation resources to complete the CDC tasks of the vehicles. Hence, even by submitting a larger buy-bid, the buyers may still not be allocated any seller for their CDC tasks.

In Fig. 2, we observe that sellers with sell-bids of larger than 122 are not allocated to any buyer and thus

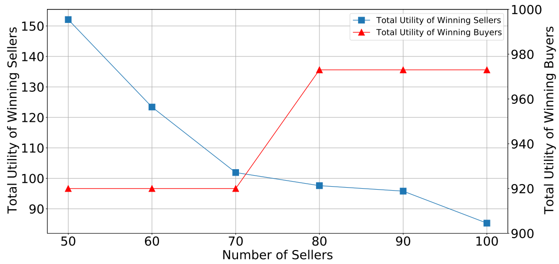

Figure 3: Utilities of winning sellers and buyers under different number of sellers and 6 buyers.

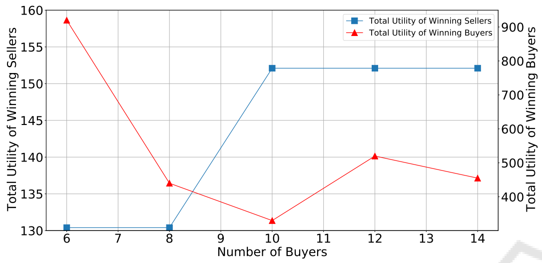

Figure 4: Utilities of winning sellers and buyers under different number of buyers and 50 sellers.

their utilities are zero. The reasons are similar to those of the buyers. Firstly, the buyers that can be matched to the sellers need to have larger buy-bids than the sellers' sell-bids. This reduces the number of possible buyers that can be matched to the sellers with large sell-bids. Secondly, the sellers with large sell-bids lose out to those with small sell-bids. We also observe that sellers with smaller sell-bids, e.g., 14.8 and 45.2, are also not allocated to any buyer. This is because they may not have sufficient resources to complete the CDC subtasks of the buyers. For sellers that have sufficient resources for the CDC subtasks of the buyers, the match between the seller and buyer is still not successful if the number of possible sellers that can be matched to the buyers does not meets the buyer's recovery threshold requirement.

Given that the double auction mechanism is individually-rational and incentive-compatible, we study the effect of different number of buyers and sellers on their utilities as well as the utility of the auctioneer. Figure 3 shows the utilities of winning buyers and sellers in a distributed vehicular edge computing network that consists of 6 buyers and different number of sellers. As the number of sellers increases, the total utility of winning sellers decreases. Since the competition among the sellers increases, the sellers submit lower sell-bids to the auctioneer, resulting in lower selling prices. Hence, the utilities of the winning sellers decrease. However, the buyers benefit from an increase in number of sellers. We observe that as the number of sellers increase from 70 to 80, the total utility of the winning buyers increases from 920 to 973. With a larger number of sellers in the network, more buyers are able to be matched with the sellers and win the buy-bids, hence completing their CDC tasks and receiving positive utilities.

Similar trend can be observe when the number of buyers varies. In Fig. 4, as the number of buyers increases, the total utility of winning buyers generally decreases. The increase in competition between the buyers results in larger buy-bids submitted by the buyers and hence, higher payment prices for the resources of the sellers. As such, the utilities of the winning buyers decrease. However, the total utility of the winning sellers increases. As the number of buyers increases, there is a larger demand for the resources of the sellers and more sellers are allocated to facilitate the CDC tasks of the buyers. Thus, the total utility of the winning sellers increases.

5 CONCLUSION

In this paper, we proposed a double auction mechanism to incentivize the edge servers to facilitate the vehicles in completing their CDC tasks. Firstly, the vehicles use PolyDot codes to split their CDC tasks and distribute the CDC subtasks to the edge servers for computations. Then, based on the requirements of the vehicles, the double auction mechanism matches the vehicles and edge servers as well as determines the payment prices of the vehicles and selling prices of the edge servers. For our future work, we can consider different types of distributed computation tasks, e.g., gradient descent, convolution and Fourier Transform.

ACKNOWLEDGEMENTS

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), and Singapore Energy Market Authority (EMA), Energy Resilience, under Grant NRF2017EWT-EP003-041. This research is also supported by WASP/NTU grant M4082187 (4080) and Singapore Ministry of Education (MOE) Tier 1 (RG16/20).

REFERENCES

Dutta, S., Fahim, M., Haddadpour, F., Jeong, H., Cadambe, V., and Grover, P. (2020). On the Optimal Recovery Threshold of Coded Matrix Multiplication. *IEEE Transactions on Information Theory*, 66(1):278–301.

Fusco, G., Colombaroni, C., Comelli, L., and Isaenko, N. (2015). Short-term Traffic Predictions on Large Urban Traffic Networks: Applications of Network-based Machine Learning Models and Dynamic Traffic Assignment Models. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 93–101, Budapest, Hungary.

Lin, H., Lin, Y., and Chang, W. (2012). Reputation Auction Framework for Cooperative Communications in Green Wireless Networks. In *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pages 875–880, Sydney, New South Wales, Australia.

Liu, Y., Zhang, J., Yu, H., and Miao, C. (2014). Reputation-aware Continuous Double Auction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).

Radu, M. D., Costea, I. M., and Stan, V. A. (2020). Automatic Traffic Sign Recognition Artificial Intelligence - Deep Learning Algorithm. In *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–4, Bucharest, Romania.

Xu, X., Xue, Y., Li, X., Qi, L., and Wan, S. (2019). A Computation Offloading Method for Edge Computing With Vehicle-to-Everything. *IEEE Access*, 7:131068–131077.

Yu, Q., Maddah-Ali, M., and Avestimehr, S. (2017). Polynomial Codes: an Optimal Design for High-Dimensional Coded Matrix Multiplication. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 4403–4413, Long Beach, California, USA. Curran Associates, Inc.