End-to-End Service Auction: A General Double Auction Mechanism for Edge Computing Services

Xianhao Chen, *Student Member, IEEE*, Guangyu Zhu, Haichuan Ding, Lan Zhang, *Member, IEEE*, Haixia Zhang, *Senior Member, IEEE*, and Yuguang Fang, *Fellow, IEEE*

***Abstract*— Ubiquitous powerful personal computing facilities, such as desktop computers and parked autonomous cars, can function as micro edge computing servers by leveraging their spare resources. However, to harvest their resources for service provisioning, two significant challenges will arise: how to incentivize the server owners to contribute their computing resources, and how to guarantee the end-to-end (E2E) Quality-of-Service (QoS) for service buyers? In this paper, we address these two problems in a holistic way by advocating COMSA. Unlike the existing double auction schemes for edge computing which mostly focus on computing resource trading, COMSA addresses the joint problem of double auction mechanism design and network resource allocation by explicitly taking spectrum allocation and data routing into account, thereby providing E2E QoS guarantees for edge computing services. To handle the design complexity, COMSA employs a two-step procedure to decouple network optimization and mechanism design, which hence can be applied to general network optimization problems for edge computing. COMSA holds some critical economic properties, i.e., truthfulness, budget balance, and individual rationality. Our extensive simulation studies demonstrate the effectiveness of COMSA.**

***Index Terms*— Edge computing, double auction, spectrum allocation, service provisioning.**

I. Introduction

THE proliferation of Internet of Things (IoT) and various mobile devices has given birth to a broad set of applications, such as video analytics, environmental monitoring, virtual/augmented reality, and online gaming [1]. These newly emerging services induce soaring demands for data analytics and intelligence extraction. To accommodate the service demands, multi-access edge computing (MEC) has been conceived to provide powerful computing capabilities in the close proximity of end devices [1], [2]. Benefiting from the short distance between data sources and servers, MEC can support various data analytics applications with low latency and high throughput.

The ever-increasing user demands call for dense deployment of powerful edge servers within the network edge [3], particularly in populated areas. However, due to the installation cost, ubiquitous deployment of edge servers is unrealistic at least in the near future. At the same time, we have witnessed that various privately-owned facilities, such as personal computers and autonomous cars [4]–[6], are being equipped with significantly powerful computing capabilities. To address the aforementioned dilemma, an intuitive and promising solution is to harvest the abundant computing resources of these personal devices when they are idle, e.g., when autonomous cars are parked, to beef up the existing computing infrastructure for service provisioning.

While it is absolutely compelling to leverage these personal facilities as edge servers, two fundamental problems should be addressed before pushing it into reality: how to stimulate the server owners to contribute their computing resources, and how to guarantee the service subscribers to get high-quality end-to-end (E2E) services? To address the first problem, auction is a natural choice to provide the needed incentives to the selfish agents because of its efficiency in bridging supplies and demands without needing prior information about the agents' valuations [7]. The second problem stems from the fact that MEC service provisioning requires the appropriate management of both communication and computing resources. Taking video analytics as an example, the real-time video processing on edge servers requires not only considerable computing power, but also high bandwidth from end devices to edge servers for video transmissions. As such, an infrastructure should judiciously allocate not only the harvested computing resource, but also limited network resource, e.g., spectrum, to support MEC services with E2E QoS guarantees. Although some prior works have sought to design double auction mechanisms for MEC [8], [9], they mostly focus on the aspect of computing resource trading while implicitly assuming that communication resources are sufficient. For this reason, the auction could fail to meet the E2E QoS demands from winning service requests in a practical network, and service buyers may get charged even if they do not actually benefit from the trades due to network congestion.

Manuscript received April 12, 2021; revised November 11, 2021 and February 23, 2022; accepted May 9, 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. Iosifidis. This work was supported in part by the U.S. National Science Foundation under Grant IIS-1722791 and Grant CNS-2106589. The work of Haichuan Ding was supported in part by the Young Elite Scientists Sponsorship Program by CIC under Grant 2021QNRC001. The work of Haixia Zhang was supported in part by the Project of International Cooperation and Exchanges NSFC under Grant 61860206005. *(Corresponding author: Haixia Zhang.)*

Xianhao Chen, Guangyu Zhu, and Yuguang Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: xcheneee@hku.hk; gzhu@ufl.edu; fang@ece.ufl.edu).

Haichuan Ding is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (email: hcding@bit.edu.cn).

Lan Zhang is with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931 USA (e-mail: lanzhang@mtu.edu).

Haixia Zhang is with the Shandong Provincial Key Laboratory of Wireless Communication Technologies, Shandong University, Jinan 250061, China (e-mail: haixia.zhang@sdu.edu.cn).

Digital Object Identifier 10.1109/TNET.2022.3179239

Motivated by the above observations, we propose COMputing Service Auction (COMSA) for MEC. Specifically, we study the joint problem of double auction and E2E service provisioning for MEC systems, where the central controller, called computing service provider (CSP), not only harvests available computing resource via auction, but also guarantees the necessary E2E data transmissions between winning buyers and sellers. In practice, CSP can be part of a mobile network operator which owns infrastructure and spectrum bands, but lacks computing resource, thus establishing a double auction to incentivize computing server owners to contribute. The name "E2E service auction" indicates that our auction mechanism is "service-oriented", which *treats E2E edge computing services as commodities* by jointly allocating network and computing resources to winning buyers to satisfy their E2E QoS requirements. On the contrary, the existing double auction mechanisms for MEC mostly focus on the computing aspect. In particular, while spectrum allocation and data routing form the basis for E2E QoS guarantees for edge computing services, they have not been considered in the existing double auction mechanisms for MEC systems. For the high-level ideas and broader applications of E2E service auction, please refer to our tutorial paper [10] for details.

The key contributions of this paper are summarized as follows.

- To provide E2E QoS guarantees for service buyers, we propose the first truthful service-oriented double auction mechanism for MEC which takes network optimization, particularly spectrum allocation and data routing, into consideration. To tackle the design complexity, we propose a two-step approach to decouple network optimization and mechanism design, which can be applied to general network optimization problems for MEC.
- In our framework, each buyer is allowed to initiate multiple requests, and each seller is allowed to serve multiple requests. We present a novel pair partitioning method to address the truthful auction mechanism design. Allowing each buyer to initiate multiple service requests also distinguishes COMSA from the existing truthful double auction schemes for MEC.
- Our proposed mechanism is shown to be truthful, individually rational, and budget balanced. Extensive simulations demonstrate that our scheme can guarantee the truthfulness without resulting in significant performance degradation.

II. Related Work

Regarding the resource allocation and computing offloading for MEC, most early works study the case of a single server with a single/multiple users [11]–[13]. In light of the limited capacity of edge servers and their dense deployment, a few recent works further consider resource provisioning among a number of geo-distributed edge servers [3], [14]–[16]. In [3], Ding *et al.* study the placement of MEC services in wireless mesh networks by considering both spectrum allocation and data routing, where the relay nodes equipped with cognitive radios deliver the input data from data sources to the assigned edge servers.

To create incentives for computing server owners and provide services to end users, double auction is an effective approach that naturally fits this double-sided market. Our double auction mechanism trades "end-to-end services" instead of pure computing resource to end users, thus requiring the joint management of computing and network (e.g. spectrum) resource. Essentially, our double auction mechanism trades heterogeneous items (service requests) between buyers and sellers, under networking constraints on the set of buyers and sellers that can trade with each other. This stems from the fact that the set of winning requests have to be supported simultaneously under E2E QoS constraints over a resource-limited wireless network. In economic literature, McAfee presents a truthful budget-balanced double auction mechanism where buyers and sellers exchange homogeneous items in the seminal work [17]. After that, there are a few works on truthful multi-unit double auction with single or multiple good types [18]–[21]. However, the aforementioned works do not impose constraints on the trades. In [22], Dütting *et al.* develop a modular approach to designing double auction schemes with social welfare approximation guarantee under a constraint on trades. However, their scheme considers homogeneous items, which cannot be applied to our case either.

When applying double auctions to wireless networks, there are inherent wireless interference constraints limiting the set of buyers and sellers that can trade with each other. Along this line, double auctions with heterogeneous items have been investigated for dynamic spectrum trading [23]–[25] and cooperative or device-to-device communications [26], [27]. This kind of mechanisms contain two steps: a bid-independent assignment process and a winner determination and pricing process. Enlightened by this line of research, our COMSA mechanism adopts such a two-step process, while having variants in both steps. In the first step, COMSA involves solving a network resource optimization (e.g., a joint spectrum allocation and request routing) problem, unlike the existing auction schemes which only focus on buyer-seller assignment. In the second step, COMSA adopts a novel pair partitioning strategy to tackle many-to-many buyer-seller assignment. In fact, by simple modification, a two-step double auction for cooperative communications can be applied to our case [26]. Unfortunately, their scheme is designed for one-to-one buyer-seller assignment, which could underutilize the resources of edge servers.

In the context of cloud/edge computing, some single-sided auction schemes have been proposed [28]–[31], which only consider either bid prices from service buyers or ask prices from server owners. Unfortunately, the former case does not consider the incentive issue for server owners, while in the latter case, it is unclear how to charge buyers properly and even who and where the service buyers are. Similar to our work, some truthful double auction mechanisms have been presented to incentivize individual cloud/edge servers [8], [9], [32]–[34]. Nonetheless, the aforementioned works do not consider network resource allocation, particularly spectrum allocation and data routing, in their mechanism design.

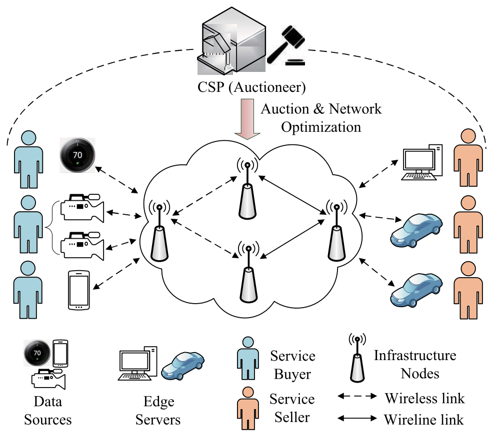


Fig. 1. An exemplary paradigm of COMSA.

The only exception is a recent work on double auction mechanism design for MEC systems, which considers band allocation [35]. However, in their scheme, every edge server is assumed to hold independent spectrum resource, which prevents spectrum reuse among edge servers and might not be realistic for dense wireless networks. Also, data routing has not been investigated therein. Considering the fact that there are multiple servers (sellers) in the double auction market, appropriate spectrum allocation (reuse) is crucial to mitigate signal interference and improve resource utilization, while judicious data routing is essential for flow balancing and E2E QoS guarantees if considering multi-hop networks.

III. System Model

A. Network Architecture

As shown in Fig. 1, there are three parties in our system:

Computing Service Provider (CSP) provides edge computing services to service buyers by leveraging the computing resources from sellers. CSP owns some necessary infrastructure and radio resources, such as (small) base stations, relays, and spectrum bands, to support the data delivery between data sources and edge servers.

Service Sellers are the server owners. We assume that each seller $j \in \mathcal{J}$ has one server, which may host multiple computing services simultaneously according to its resource availability. The state of the server of seller j is described as a tuple $\{\Theta_j, \Phi_j\}$, where Θ_j represents the available computing capability, and Φ_j represents the available memory space.

Service Buyers are the users who send the computing service requests to CSP. The input data of each service request is assumed to be generated from a source node, and needs to be delivered to an edge server for processing or computing. We assume that each buyer $i \in \mathcal{I}$ initiates $K_i \geq 1$ service requests, implying that it has K_i source nodes. The set of QoS requirements for the k-th service request of buyer i is described by a tuple $S_{i,k} = \{r_{i,k}, \theta_{i,k}, \phi_{i,k}\}$, where $r_{i,k}$, $\theta_{i,k}$, and $\phi_{i,k}$ denote the E2E data rate requirement (in bps), the computing requirement (in Hz), and the memory requirement (in Byte), respectively. It is worth noting that the service requests with QoS requirements $S_{i,k}$ are exactly the commodities to be traded in our scheme.

For example, suppose that a surveillance camera generates continuous video stream with the resolution of 1280×720 and frame rate $30fps$, which needs to be processed at the edge. This request can be naturally mapped to a set of QoS requirements. When H.264 is used for video compression, the request demands E2E data rate 2Mbps [36]. Assuming 24 bits for the RGB color values of a pixel, if the adopted algorithm needs 5 CPU cycles to process one bit, it requires the processing power of 3.3GHz. Also, assume that the request demands 1GB memory. Overall, the corresponding QoS requirements are $r_{i,k} = $ 2Mbps, $\theta_{i,k} = $ 3.3GHz, and $\phi_{i,k} = $ 1GB.

B. Double Auction Model

Next, we illustrate the system in Fig. 1 from an economic perspective. We characterize the interactions between service buyers and service sellers as a single-round sealed-bid double auction, where CSP serves as the auctioneer. Throughout this paper, we also call both buyers and sellers as agents. CSP conducts the auction and network optimization at the beginning of each period, and we only need to focus on one period.[1] Both computing service requests and sellers' computing servers are active during the considered period. Let $\boldsymbol{V}_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,K_i}\}$ be the true valuation vector of buyer i, where $v_{i,k}$ denotes the true valuation of buyer i towards its k-th service request,[2] which describes the maximum price at which the buyer is willing to pay for the request. Note that a buyer has no preference towards different edge servers, because it is only concerned about whether the QoS requirements of its request are met. Likewise, let $\boldsymbol{C}_j = \{c_{1,j,1}, c_{1,j,2}, \ldots, c_{|\mathcal{I}|,j,K_i}\}$ be the true cost vector of seller j, in which $c_{i,j,k}$ denotes the true cost of seller j for serving buyer i's k-th service request. For seller j, its true cost depends on the resource consumption on its server. Given the fact that the service requests may demand different amount of resource, a seller has potentially different true costs for them.

At the beginning of the auction, buyer i offers a bid vector $\boldsymbol{B}_i = \{b_{i,1}, b_{i,2}, \ldots, b_{i,K_i}\}$ to CSP, where $b_{i,k}$ represents buyer i's bid price for its k-th service request. Since $\boldsymbol{V}_i$ is privately known by buyer i, $\boldsymbol{B}_i$ may not be equal to $\boldsymbol{V}_i$. The utility of buyer i is the total valuation for its winning services minus its total payment:

$$U_i^b = \sum_{1 \leq k \leq K_i} y_{i,k}^b (v_{i,k} - \overline{b}_{i,k}), \quad \forall i \in \mathcal{I}, \tag{1}$$

where $y_{i,k}^b = 1$ represents that buyer i wins its k-th service request, and $y_{i,k}^b = 0$ otherwise; $\overline{b}_{i,k}$ is the clearing price for buyer i in terms of its k-th request, which will be determined by our pricing policy. Besides, we use $U_{i,k}^b = y_{i,k}^b (v_{i,k} - \overline{b}_{i,k})$ to denote the utility that buyer i gains from its k-th request.

Similarly, seller j offers an ask vector $\boldsymbol{A}_j = \{a_{1,j,1}, a_{1,j,2}, \ldots, a_{|\mathcal{I}|,j,K_i}\}$ to CSP, where $a_{i,j,k}$

[1] Once the auction closes, the resource allocation will not be changed over the whole optimization period. The optimization period depends on the service type, network scale, and channel state. In general, in a more stationary network, the optimization period can be longer, and vice versa.

[2] Another direction is to consider all the requests from a buyer as a single bid. However, in this model, the buyer can only win either all of them or nothing, thus decreasing buyers' opportunities to win.

represents its ask price for buyer i's k-th service request. Again, $\boldsymbol{A}_j$ may not be equal to $\boldsymbol{C}_j$, as $\boldsymbol{C}_j$ is privately known by seller j. The utility of seller j is equal to its total reward minus its total cost for its winning (serving) services:

$$U_j^s = \sum_{i \in \mathcal{I}} \sum_{1 \leq k \leq K_i} y_{i,j,k}^s (\overline{a}_{i,j,k} - c_{i,j,k}), \quad \forall j \in \mathcal{J}, \tag{2}$$

where $y_{i,j,k}^s = 1$ represents that seller j wins buyer i's k-th request, and $y_{i,j,k}^s = 0$ otherwise; $\overline{a}_{i,j,k}$ is the clearing price for seller j in terms of buyer i's k-th request. Moreover, we use $U_{i,j,k}^s = y_{i,j,k}^s(\overline{a}_{i,j,k} - c_{i,j,k})$ to denote the utility that seller j gains from buyer i's k-th request.

Given bid vectors $\boldsymbol{B}_i$ for $i \in \mathcal{I}$, ask vectors $\boldsymbol{A}_i$ for $j \in \mathcal{J}$, and the network conditions, CSP should allocate the network resources, make the service assignment decisions (i.e., selecting winning bids and asks), and determine the clearing prices for both sides. The utility of CSP (auctioneer) is equal to the total service charges collected from the buyers, minus the expense paid to the sellers:

$$U^A = \sum_{i \in \mathcal{I}} \sum_{1 \leq k \leq K_i} y_{i,k}^b \overline{b}_{i,k} - \sum_{i \in \mathcal{I}} \sum_{1 \leq k \leq K_i} \sum_{J \in \mathcal{J}} y_{i,j,k}^s \overline{a}_{i,j,k}. \tag{3}$$

In our system, the auction mechanism aims to (approximately) maximize the system efficiency, which is defined as the weighted sum of accepted services in this paper (see Section IV for details). We assume that CSP is trustworthy, i.e., executing the auction mechanism faithfully, and therefore it does not attempt to maximize U^A in (3) while just ensuring $U^A \geq 0$ to avoid a deficit [23], [25], [26]. This is a reasonable assumption, since CSP, like a wireless service provider, has the motivation to maintain its reputation and service acceptance ratio. For readers' convenience, the frequently used notations in this paper are summarized in Table I.

TABLE I

FREQUENTLY USED NOTATIONS

Notation	Description
$\mathcal{I}$	The set of buyers
K_i	The number of requests from buyer i
$\mathcal{J}$	The set of sellers
$\mathcal{N}_s$	The set of source nodes
$\mathcal{N}_r$	The set of infrastructure nodes
$\mathcal{N}_d$	The set of server nodes
$\mathcal{N}$	The set of overall network nodes
$\mathcal{L}$	The set of links in the network
$r_{i,k}$	The E2E data rate requirement for buyer i's k-th service
$\theta_{i,k}$	The computing requirement for buyer i's k-th service
$\phi_{i,k}$	The storage requirement for buyer i's k-th service
$\boldsymbol{V}_i$	The valuation vector of buyer i
$\boldsymbol{B}_i$	The bid vector of buyer i
$b_{i,k}/v_{i,k}$	Buyer i's bid price/true valuation for its k-th service request
$\overline{b}_{i,k}$	The clearing price for bid $b_{i,k}$
$\boldsymbol{C}_j$	The cost vector of seller j
$\boldsymbol{A}_j$	The ask vector of seller j
$a_{i,j,k}/c_{i,j,k}$	Seller j's ask price/true cost for buyer i's k-th service request
$\overline{a}_{i,j,k}$	The clearing price for ask $a_{i,j,k}$
U_i^b	The utility of buyer i
U_j^s	The utility of seller j
U^A	The utility of CSP (auctioneer)
$d_{i,k}^j$	Equals 1 if buyer i's k-th service is assigned to seller j and equals 0 otherwise

C. Desirable Economic Properties

As in [23], [25], [26], our double auction mechanism should satisfy the following economic properties:

- *Budget Balance:* A double auction is budget balanced if the payment charged from buyers is no less than the payment paid to the sellers, i.e., $U^A \geq 0$.
- *Individual Rationality:* A double auction is individually rational if no buyer pays more than its bid price for each winning bid, and no seller is paid less than its ask price for each winning ask,[3] i.e., $\overline{b}_{i,k} \leq b_{i,k}$ and $\overline{a}_{i,j,k} \geq a_{i,j,k}$.
- *Truthfulness:* In a double auction, truthfulness means that no buyer/seller can improve its utility by claiming a bid/ask price deviating from its true valuation/cost. In other words, the optimal strategy for buyer i is to bid $\boldsymbol{B}_i = \boldsymbol{V}_i$, and the optimal strategy for seller j is to ask $\boldsymbol{A}_j = \boldsymbol{C}_j$, no matter how other agents bid/ask.

Budget balance, individual rationality, and truthfulness are three critical economic properties for a double auction mechanism. Budget balance guarantees that there is no deficit for the auctioneer. Individual rationality encourages service buyers to initiate service requests, and incentivizes sellers to contribute their resources. Truthfulness eliminates the expensive overhead of agents for strategizing over others. From agents' perspectives, such cost is pure waste which decreases their utilities. From the system perspective, the complex strategization and the fear of market manipulation may discourage many users with less powerful resources from participation, thus producing very poor auction outcome. This is especially the case in the context of wireless networks, where end users generally want to acquire services from a user-friendly platform with simple operations. Also, since buyers have no incentives to bid lower than their true valuations and sellers have no incentives to ask higher than their true costs, truthfulness can potentially increase the number of trades under the budget balance requirement. Due to the importance of these three economic properties, we aim to achieve them first while approximately maximizing the system efficiency, which is also a widely-adopted design objective in double auction [23], [25], [26].

[3]Our defined individual rationality is at bid/ask level, which is a stronger definition than the individual rationality at agent level. Since one agent can propose multiple bids/asks in our system, only guaranteeing individual rationality at agent level may discourage an agent from proposing some bids/asks.

IV. GENERAL SERVICE PROVISIONING PROBLEM

The main challenge in designing COMSA is that, the truthful double auction design, i.e., winner determination and pricing, is intrinsically coupled with the network optimization. For a single-side auction model, one may adapt the well-known Vickrey-Clarke-Groves (VCG) scheme to tackle this kind of joint problem as done in [37]. Unfortunately, the VCG-style double auction violates budget balance. To make the considered problem tractable, we devise a two-step procedure to first obtain the "candidate" service assignment result from a bid-independent service provisioning (ISP) problem which does not take bid and ask prices into account, and then make winner determination and pricing for the candidate services to achieve the desirable economic properties. The ISP problem avoids the appearance of prices so that buyers and sellers are unable to affect its solution via market manipulation. To improve interpretability, we first present a general formulation for the ISP problem in this section, and then get into the auction mechanism design in the next section.

Let us consider an abstract communication network as illustrated in Fig. 1, where a set $\mathcal{N}_s$ of source nodes and a set $\mathcal{N}_d$ of server (destination) nodes are at fixed positions. CSP employs a set $\mathcal{N}_r$ of relay nodes and a set of bands (e.g., cellular bands) to facilitate data transmissions between these two sides. We denote by $\mathcal{L}$ the set of *transmission* links in the network, where a wired link $(m,n) \in \mathcal{L}$ exists only if there is a direct wireline connection from node m to node n, while a wireless link $(m,n) \in \mathcal{L}$ exists only if the received power at node n is greater than a threshold when the transmitter m uses the maximum transmit power. Formally, the network can be represented by a directed graph $G = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the overall set of nodes in the network. We define a $|\mathcal{N}| \times |\mathcal{L}|$ incidence matrix $\boldsymbol{A}$ of the graph such that entry $a_{m,l} = 1$ if m is the transmitter of link l, -1 if m is the receiver of link l, and 0 otherwise. We use $\boldsymbol{s}_{i,k}$ to denote a vector where the m-th entry $s_m^{i,k} = 1$ if m is the source node of user i's k-th request and $s_m^{i,k} = 0$ otherwise, and $\boldsymbol{h}_j$ to denote a vector where the m-th entry $h_m^j = 1$ if m is the server of seller j and $h_m^j = 0$ otherwise.

We consider the following decision variables. Let $d_{i,k}^j \in \{0,1\}$ be the service assignment variable, where $d_{i,k}^j = 1$ only when the k-th service of buyer i is assigned to seller j. Moreover, we use flow allocation vector $\boldsymbol{f}_{i,k} = [f_1^{i,k} \dots f_{|\mathcal{L}|}^{i,k}]$ to denote the flow rate for user i's k-th request over link l, which is a non-negative vector. In addition, let $\boldsymbol{x}$ be the collection of network resource allocation variables, such as transmit power allocation and/or spectrum allocation variables. We formulate a general ISP problem as follows

$$\textbf{P1:} \quad \max_{\boldsymbol{d},\boldsymbol{x},\boldsymbol{f}} \sum_{i\in\mathcal{I}} \sum_{1\le k\le K_i} \sum_{j\in\mathcal{J}} M_{i,k} d_{i,k}^j,$$

$$s.t. \sum_{j\in\mathcal{J}} d_{i,k}^j \le 1, \quad \forall i \in \mathcal{I}, 1 \le k \le K_i, \tag{4}$$

$$\boldsymbol{A}\boldsymbol{f}_{i,k}^{\mathsf{T}} = \sum_{j\in\mathcal{J}} d_{i,k}^j r_{i,k} (\boldsymbol{s}_{i,k} - \boldsymbol{h}_j)^{\mathsf{T}}, \quad \forall i \in \mathcal{I}, 1 \le k \le K_i, \tag{5}$$

$$\sum_{i\in\mathcal{I}} \sum_{1\le k\le K_i} f_l^{i,k} \le C_l(\boldsymbol{x}), \quad \forall l \in \mathcal{L}, \tag{6}$$

$$\sum_{i\in\mathcal{I}} \sum_{1\le k\le K_i} d_{i,k}^j \theta_{i,k} \le \Theta_j, \quad \forall j \in \mathcal{J}, \tag{7}$$

$$\sum_{i\in\mathcal{I}} \sum_{1\le k\le K_i} d_{i,k}^j \phi_{i,k} \le \Phi_j, \quad \forall j \in \mathcal{J}, \tag{8}$$

$$f_l^{i,k} \ge 0, \quad \forall i \in \mathcal{I}, 1 \le k \le K_i, l \in \mathcal{L}, \tag{9}$$

$$d_{i,k}^j \in \{0,1\}, \quad \forall i \in \mathcal{I}, 1 \le k \le K_i, j \in \mathcal{J}, \tag{10}$$

$$\text{Constraints on } \ \boldsymbol{x}, \tag{11}$$

where $\boldsymbol{d}$ and $\boldsymbol{f}$ are the collections of variables $d_{i,k}^j$ and $f_l^{i,k}$, respectively, and $M_{i,k}$ is the weighting factor for each service. In particular, when $M_{i,k} = r_{i,k}$, the objective is to maximize the system service throughput (processed bits per second). When $M_{i,k} = 1$ for all services, the objective is to maximize the number of accepted services. (4) indicates that a service request is served by at most one edge server. (5) is the flow conservation equations for data routing [38], which ensures that the E2E data rate $r_{i,k}$ of buyer i's k-th service can be supported.[4] (6) means that the aggregated data flow over each link should not exceed the link capacity, where the capacity of link l, i.e., $C_l(\boldsymbol{x})$, is taken over the network resource allocation decision $\boldsymbol{x}$. (7) and (8) ensure that each server should have enough computing and memory resources to support the assigned service requests. (11) represents the general constraints for $\boldsymbol{x}$. To show the broad applicability of our design, $\boldsymbol{x}$ and $C_l(\boldsymbol{x})$ are not specified so far. A concrete **P1** with spectrum allocation variables $\boldsymbol{x}$ will be presented in Section VI.

In addition, in view of the fact that buyers and sellers may not be willing to relay data for others unless there are extra incentives to them, we can add the following constraints to ensure that they are not employed as relays.

$$\sum_{l\in\boldsymbol{L}_{n_{i,k}}^r} \sum_{i'\in\mathcal{I}} \sum_{1\le k'\le K_{i'}} f_l^{i',k'} = 0, \quad \forall i \in \mathcal{I}, 1 \le k \le K_i, \tag{12}$$

$$\sum_{l\in\boldsymbol{L}_{n_j}^t} \sum_{i\in\mathcal{I}} \sum_{1\le k\le K_i} f_l^{i,k} = 0, \quad \forall j \in \mathcal{J}, \tag{13}$$

where $\boldsymbol{L}_n^t$ or $\boldsymbol{L}_n^r$ denote the set of links whose transmitter or receiver is node n, respectively. (12) and (13) mean that there is no incoming flow of the source node of buyer i's k-th request, denoted by $n_{i,k} \in \mathcal{N}_s$, and no outgoing flow of the edge server of seller j, denoted by $n_j \in \mathcal{N}_d$, respectively.

V. COMPUTING SERVICE AUCTION (COMSA) MECHANISM

In this section, we present the COMSA mechanism, which consists of two steps: 1) bid-independent optimization & pair partitioning, 2) winning pair determination & pricing.

A. Step One: Bid-Independent Optimization & Pair Partitioning

In this step, we first obtain the service assignment results from a bid-independent optimization problem, and then divide these results into three subsets for further operations. We first

[4]As in done [3], [14], we only consider the input data stream, while not explicitly considering the computing results that edge servers send back, as the size of result is typically much smaller than the input data size and hence easy to handle.

reformulate Problem **P1** by introducing an additional constraint (14):

$$\textbf{P2:} \quad \max_{\boldsymbol{d},\boldsymbol{x},\boldsymbol{f}} \sum_{i\in\mathcal{I}} \sum_{1\le k\le K_i} \sum_{j\in\mathcal{J}} M_{i,k} d_{i,k}^{j},$$
$$s.t. \ (4)-(13),$$
$$\sum_{1\le k\le K_i} d_{i,k}^{j} \le 1, \quad \forall i\in\mathcal{I}, j\in\mathcal{J}, \qquad (14)$$

where Constraint (14) ensures that the service requests of the same buyer are assigned to different server nodes, which is necessary to prevent market manipulations, and the reason will be explained in Section V-B.[5] We use a set $\mathcal{P}$ of buyer-seller pairs to represent the service assignment outcome $\boldsymbol{d}$ obtained from Problem **P2**. Specifically, we characterize a buyer-seller pair by tuple $P_{i,j} = \{i, j, \kappa_{i,j}\}$, which means that buyer i's $\kappa_{i,j}$-th request is assigned to seller j according to the solution to Problem **P2**, where $\kappa_{i,j} = \sum_{1\le k\le K_i} k d_{i,k}^{j}$. Due to (14), the buyer-seller pair $P_{i,j}$ uniquely corresponds to buyer i's $\kappa_{i,j}$-th request. $P_{i,j} \in \mathcal{P}$ exists if and only if $d_{i,\kappa_{i,j}}^{j} = 1$, or equivalently, $\sum_{1\le k\le K_i} d_{i,k}^{j} = 1$, according to the solution $\boldsymbol{d}$. We call $\mathcal{P}$ as the set of candidate buyer-seller pairs, and the final winning pairs will be selected from $\mathcal{P}$ in the subsequent development.

Next, we attempt to devise a winning pair determination & pricing algorithm for $\mathcal{P}$ to guarantee the desired economic properties, particularly the truthfulness. A major difficulty in designing COMSA comes from the multi-demand nature, where each agent has multiple bids or asks. One straightforward solution is to replace an agent with multiple dummies, each proposing one bid/ask, and then apply single-demand double auctions [26] here. Unfortunately, this simple conversion leaves room for market manipulation, because a multi-demand agent can manipulate one bid/ask to affect the outcome of its other bids/asks, thereby violating the truthfulness.

To tackle the challenging multi-demand double auction mechanism design, we devise the pair partitioning procedure to strategically divide $\mathcal{P}$ into three independent subsets, which will then undergo three independent auction processes. The partitioning produces special bid-ask structures which facilitate truthful auction design. Specifically, we draw the mapping between buyers and sellers as a bipartite graph $\mathcal{G} = (\mathcal{I}, \mathcal{J}, \mathcal{E})$ (see the example in Fig. 2), where edge $(i,j) \in \mathcal{E}$ and its associated vertices denote the buyer-seller pair $P_{i,j} \in \mathcal{P}$. We construct three subgraphs $\mathcal{G}_1 = (\mathcal{I}_{\mathcal{G}_1}, \mathcal{J}_{\mathcal{G}_1}, \mathcal{E}_{\mathcal{G}_1})$, $\mathcal{G}_2 = (\mathcal{I}_{\mathcal{G}_2}, \mathcal{J}_{\mathcal{G}_2}, \mathcal{E}_{\mathcal{G}_2})$ and $\mathcal{G}_3 = (\mathcal{I}_{\mathcal{G}_3}, \mathcal{J}_{\mathcal{G}_3}, \mathcal{E}_{\mathcal{G}_3})$ of $\mathcal{G}$, whereby we partition the buyer-seller pairs into three independent subsets according to the graph structure. Concretely, $\mathcal{G}_1$ is formed by the buyer-seller pairs in $\mathcal{G}$ with sellers associated with multiple buyers. By removing edges $(i,j) \in \mathcal{E}_{\mathcal{G}_1}$ from graph $\mathcal{G}$, we obtain a new graph $\mathcal{G}' = (\mathcal{I}, \mathcal{J}, \mathcal{E}\backslash\mathcal{E}_{\mathcal{G}_1})$. Then, $\mathcal{G}_2$ is formed by the pairs in $\mathcal{G}'$ with buyers associated with multiple sellers, and $\mathcal{G}_3$ is formed by the remaining buyer-seller pairs with buyers and sellers associated with only one agent. Since each agent can be paired with either one or multiple agents, these three subgraphs cover all possible pairs in the original bipartite graph $\mathcal{G}$.[6] We use $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ to represent the set of pairs in subgraph $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$, respectively. Clearly, $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ are mutually exclusive, satisfying $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$. An example of pair partitioning is illustrated in Fig. 2, and we will continue using this instance in the subsequent development.

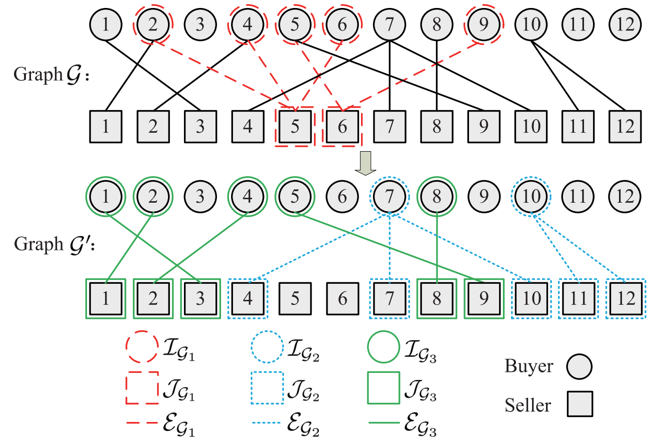


Fig. 2. Illustration of buyer-seller pair partitioning. Graph $\mathcal{G} = (\mathcal{I}, \mathcal{J}, \mathcal{E})$ is partitioned into three subgraphs, i.e.,$\mathcal{G}_1 = (\mathcal{I}_{\mathcal{G}_1}, \mathcal{J}_{\mathcal{G}_1}, \mathcal{E}_{\mathcal{G}_1})$, $\mathcal{G}_2 = (\mathcal{I}_{\mathcal{G}_2}, \mathcal{J}_{\mathcal{G}_2}, \mathcal{E}_{\mathcal{G}_2})$, and $\mathcal{G}_3 = (\mathcal{I}_{\mathcal{G}_3}, \mathcal{J}_{\mathcal{G}_3}, \mathcal{E}_{\mathcal{G}_3})$.

Note that step one is independent of bid/ask prices. Thus, any agent is unable to manipulate their bids/asks to impact the pair partitioning results, i.e., which subgraph their bids/asks would enter. Moreover, in the following mechanism design, the winning result and pricing for a bid/ask only hinge on the subgraph that it enters. Consequently, designing a truthful winning pair determination and pricing mechanism for $\mathcal{P}$ is boiled down to designing truthful mechanisms for subset $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$, respectively.

B. Step Two: Winning Pair Determination & Pricing

In this step, we present three winning pair determination & pricing schemes, namely Subroutine1, Subroutine2, and Subroutine3, for $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, respectively. To ensure truthfulness by considering the multi-demand nature, our basic idea is to eliminate some pairs and set the clearing prices to the removed bids/asks or certain thresholds (defined later), so that the fate of each bid/ask cannot be affected by other bids/asks proposed by the same agent.

In our design, CSP introduces thresholds $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\max}$ to ensure that all the effective bids fall in the range of $[b_{\text{th}}^{\min}, +\infty)$, and all the effective asks fall in the range of $(0, a_{\text{th}}^{\max}]$. The bids

[5]The additional constraint (14) may make the optimal solution to Problem **P2** inferior to Problem **P1** due to the decrease in assignment flexibility. However, this impact would be small in many realistic scenarios. For instance, a multi-request buyer owning multiple data sources (such as surveillance cameras) usually deploys them at different locations, which are easy to be associated with different servers. Moreover, in the special case where each buyer only has one request, (14) can be removed.

[6]One can change the order of constructing $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$. The desirable economic properties can still be guaranteed, because each agent cannot control the pair partitioning procedure no matter which construction order is used. However, since the buyer-seller pairs appear in the previous subgraph(s) will not appear in the later subgraph(s), the final auction result may vary across different construction orders.

and asks out of the above intervals will be directly rejected.[7] In practice, $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\max}$ can be determined by the auctioneer based on the historical information. Notice that COMSA is still applicable to the cases where appropriate $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\max}$ are hard to obtain due to the lack of prior information, for which the auctioneer can simply set $b_{\text{th}}^{\min}$ to 0 and $a_{\text{th}}^{\max}$ to $+\infty$. However, we will show later that one can significantly boost the system performance by employing proper $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\max}$.

1) Subroutine1: Subroutine1 is designed for $\mathcal{P}_1$. As illustrated in Fig. 3, graph $\mathcal{G}_1$ is a forest formed by multiple trees, where each seller is the root of each tree. The winner determination and pricing are conducted for each tree independently. We define $\mathcal{G}_1^j = \{\mathcal{I}_{\mathcal{G}_1^j}, \{j\}, \mathcal{E}_{\mathcal{G}_1^j}\}$ as the tree whose root is seller $j \in \mathcal{J}_{\mathcal{G}_1}$. Let $s_j = \arg\min_{i \in \mathcal{I}_{\mathcal{G}_1^j}} b_{i,\kappa_{i,j}}$ be the buyer index with the lowest bid price in tree $\mathcal{G}_1^j$. If $b_{s_j,\kappa_{s_j,j}} \geq a_{\text{th}}^{\max}$, it is known that all the sellers' ask prices are lower than or equal to the lowest bid price $b_{s_j,\kappa_{s_j,j}}$, since they are lower than or equal to $a_{\text{th}}^{\max}$. In this case, all the buyer-seller pairs in tree $\mathcal{G}_1^j$ win, and the clearing prices for all the winning bids/asks in $\mathcal{G}_1^j$ are set to $a_{\text{th}}^{\max}$. If $b_{s_j,\kappa_{s_j,j}} < a_{\text{th}}^{\max}$, the pair involving the lowest buyer, i.e., s_j, is first sacrificed. Then, the remaining pairs in tree $\mathcal{G}_1^j$ with ask prices not higher than $b_{s_j,\kappa_{s_j,j}}$ win, while the pairs with ask prices higher than $b_{s_j,\kappa_{s_j,j}}$ lose. The clearing prices for these winning bids/asks are all set to $b_{s_j,\kappa_{s_j,j}}$. We observe that the lowest buyer may not necessarily be sacrificed if $b_{s_j,\kappa_{s_j,j}} \geq a_{\text{th}}^{\max}$. As a consequence, a scheme with a proper $a_{\text{th}}^{\max}$ may perform better than that with $a_{\text{th}}^{\max} = +\infty$. We remark that $a_{\text{th}}^{\max}$ is predetermined by CSP, and cannot be affected by buyers or sellers.

Now we can explain why Constraint (14) is necessary to resist market manipulation. If we remove (14) from Problem **P2**, a tree of graph $\mathcal{G}_1$ may contain multiple bids of the same buyer. In such a case, the buyer can deliberately make one of its bids losing by changing it to the lowest bid price in this tree. Since the clearing price equals the lowest bid $b_{s_j,\kappa_{s_j,j}}$ in the case of $b_{s_j,\kappa_{s_j,j}} < a_{\text{th}}^{\max}$, the buyer can lower this value to gain more benefit for its remaining bids. A similar phenomenon can also be found in Subroutine2.

To facilitate understanding, let us consider the example in Fig. 3, where graph $\mathcal{G}_1$ is derived from Fig. 2. In this case, graph $\mathcal{G}_1$ is formed by two trees, i.e., $\mathcal{G}_1^5$ and $\mathcal{G}_1^6$. Suppose that CSP sets $a_{\text{th}}^{\max} = 4.5$. In tree $\mathcal{G}_1^5$, buyer 4 offers the lowest bid price $b_{4,\kappa_{4,5}}$. Since $b_{4,\kappa_{4,5}} < a_{\text{th}}^{\max}$, pair $P_{4,5}$ should be sacrificed. Moreover, as $a_{2,5,\kappa_{2,5}} < b_{4,\kappa_{4,5}}$ and $a_{6,5,\kappa_{6,5}} < b_{4,\kappa_{4,5}}$, pair $P_{2,5}$ and pair $P_{6,5}$ win. The clearing prices are set to $b_{4,\kappa_{4,5}}$, i.e., $\overline{b}_{2,\kappa_{2,5}} = \overline{a}_{2,5,\kappa_{2,5}} = \overline{b}_{6,\kappa_{6,5}} = \overline{a}_{6,5,\kappa_{6,5}} = 4$. In tree $\mathcal{G}_1^6$, since the lowest bid price $b_{5,\kappa_{5,6}} > a_{\text{th}}^{\max}$, both pairs in $\mathcal{G}_1^6$ win, where the clearing prices are set to $a_{\text{th}}^{\max}$, i.e., $\overline{b}_{5,\kappa_{5,6}} = \overline{a}_{5,6,\kappa_{5,6}} = \overline{b}_{9,\kappa_{9,6}} = \overline{a}_{9,6,\kappa_{9,6}} = 4.5$.

It is worthy of attention that Subroutine1 should sacrifice at least one buyer-seller pair in tree $\mathcal{G}_1^j$ if $b_{s_j,\kappa_{s_j,j}} < a_{\text{th}}^{\max}$. As a result, Subroutine1 yields no winning pair for $\mathcal{P}_2$ and $\mathcal{P}_3$ in the case of $a_{\text{th}}^{\max} = +\infty$, where each seller is only paired with

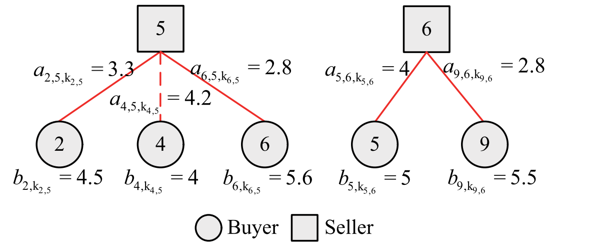


Fig. 3. An example of Subroutine1. The solid lines represent the winning pairs, and the dash line represents the losing pair.

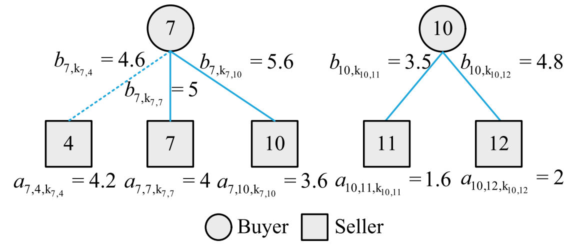


Fig. 4. An example of Subroutine2. The solid lines represent the winning pairs, and the dot line represents the losing pair.

one buyer. This observation implies that Subroutine1 cannot be applied to $\mathcal{P}_2$ and $\mathcal{P}_3$.

2) Subroutine2: Subroutine2 is developed for $\mathcal{P}_2$. Note that in the special case where each buyer only has one request, $\mathcal{P}_2$ is empty so that subroutine2 will not be executed. Subroutine2 is symmetric to Subroutine1, where the only difference is that that buyers are the roots instead (see the example in Fig. 4). We define $\mathcal{G}_2^i = \{\{i\}, \mathcal{J}_{\mathcal{G}_2^i}, \mathcal{E}_{\mathcal{G}_2^i}\}$ as the tree whose root is buyer $i \in \mathcal{I}_{\mathcal{G}_2}$. Denote t_i as the index of the seller offering the highest ask price in $\mathcal{J}_{\mathcal{G}_2^i}$, i.e., $t_i = \arg\max_{j \in \mathcal{J}_{\mathcal{G}_2^i}} a_{i,j,\kappa_{i,j}}$. If $a_{i,t,\kappa_{i,t_i}} \leq b_{\text{th}}^{\min}$, all the buyers' bid prices must be higher than or equal to the highest ask price $a_{i,t,\kappa_{i,t_i}}$. In this case, all the pairs in tree $\mathcal{G}_2^i$ win, where the clearing prices for all the bids/asks in $\mathcal{G}_2^i$ are set to $b_{\text{th}}^{\min}$. If $a_{i,t_i,\kappa_{i,t_i}} > b_{\text{th}}^{\min}$, the pair containing seller t_i is first eliminated. Among the remaining pairs in $\mathcal{G}_2^i$, the pairs with bid prices not lower than $a_{i,t_i,\kappa_{i,t_i}}$ win, whereas other pairs lose. The clearing prices for all the winning bids/asks are set to $a_{i,t_i,\kappa_{i,t_i}}$.

For clarity, Fig. 4 gives an instance of Subroutine2. Graph $\mathcal{G}_2$ is formed by two trees, i.e., $\mathcal{G}_2^7$ and $\mathcal{G}_2^{10}$. Suppose that CSP sets $b_{\text{th}}^{\min}$ to 2. In tree $\mathcal{G}_2^7$, the highest ask price $a_{7,4,\kappa_{7,4}} = 4.2 > b_{\text{th}}^{\min}$. Thus, the auctioneer sacrifices pair $P_{7,4}$. Since both $b_{7,\kappa_{7,7}}$ and $b_{7,\kappa_{7,10}}$ are higher than $a_{7,4,\kappa_{7,4}}$, $P_{7,7}$ and $P_{7,10}$ are admitted as the winning pairs, and the clearing prices are uniformly set to $a_{7,4,\kappa_{7,4}}$, i.e., $\overline{b}_{7,\kappa_{7,7}} = \overline{b}_{7,\kappa_{7,10}} = \overline{a}_{7,7,\kappa_{7,7}} = \overline{a}_{7,10,\kappa_{7,10}} = 4.2$. In tree $\mathcal{G}_2^{10}$, since the highest ask price $a_{10,12,\kappa_{10,12}} = 2 \leq b_{\text{th}}^{\min}$, both pair $P_{10,11}$ and pair $P_{10,12}$ win, with clearing price $\overline{b}_{10,\kappa_{10,11}} = \overline{b}_{10,\kappa_{10,12}} = \overline{a}_{10,11,\kappa_{10,11}} = \overline{a}_{10,12,\kappa_{10,12}} = b_{\text{th}}^{\min} = 2$.

3) Subroutine3: Subroutine3 is devised for $\mathcal{P}_3$. Since there is a simple one-to-one mapping in $\mathcal{P}_3$ (see the example in Fig. 5), we use $b_{(i)}$ and $a_{(j)}$ to represent the bid price of buyer i and the ask price of seller j in $\mathcal{P}_3$, respectively, to simplify the notation. Without loss of generality, we sort the buyers in non-increasing order in terms of their bids, i.e., $\mathbb{I} = \{i_1, i_2, i_3, \ldots, i_{|\mathcal{P}_3|}\}$, the sellers in non-decreasing

[7] In non-trivial cases, there is $a_{\text{th}}^{\max} > b_{\text{th}}^{\min}$.

order in terms of their asks, i.e., $\mathbb{J} = \{j_1, j_2, j_3 \ldots, j_{|\mathcal{P}_3|}\}$. The winning buyer-seller pairs in $\mathcal{P}_3$ is determined by a "boundary" $(\hat{x}, \hat{y})$, in the way that only the pairs containing buyer i_x that $x \leq \hat{x}$ and the sellers y that $j_y \leq \hat{y}$ win. We adopt uniform pricing policy for either side, and let B denote the clearing price for all the bids, and A denote the clearing price for all the asks. We search for the maximum g such that $b_{(i_g)} \geq a_{(j_g)}$, and then determine the boundary and clearing prices by the following conditions:

- If $g = |\mathcal{P}_3|$, we set $B = \min\{b_{(i_g)}, a_{\text{th}}^{\max}\}$, $A = \max\{a_{(j_g)}, b_{\text{th}}^{\min}\}$ and $(\hat{x}, \hat{y}) = \big(g - \mathbb{1}(b_{(i_g)} < a_{\text{th}}^{\max}), g - \mathbb{1}(a_{(j_g)} > b_{\text{th}}^{\min})\big)$, where $\mathbb{1}(\cdot)$ is an indicator function returning 1 if the condition is true and returning 0 otherwise.
- If $g < |\mathcal{P}_3|$ and $a_{(i_g)} \leq v \leq b_{(j_g)}$, we set $B = A = v$, $(\hat{x}, \hat{y}) = (g, g)$, where $v = \frac{b_{(i_{g+1})} + a_{(j_{g+1})}}{2}$. If $g < |\mathcal{P}_3|$ and $a_{(i_g)} > v$ or $v > b_{(j_g)}$, we set $B = b_{(i_g)}$, $A = a_{(j_g)}$. $(\hat{x}, \hat{y}) = (g-1, g-1)$.

From the above, we can see how the bid/ask thresholds can potentially enhance the auction performance in the case of $g = |\mathcal{P}_3|$. For the buyer side, when $b_{(i_g)} \geq a_{\text{th}}^{\max}$, B is set to $a_{\text{th}}^{\max}$; whereas if $b_{(i_g)} < a_{\text{th}}^{\max}$, B is determined by $b_{(i_g)}$. Therefore, buyer i_g can be selected as a winner in the former case while having to be sacrificed in the latter case to guarantee the truthfulness, implying that the scheme with a proper $a_{\text{th}}^{\max}$ may yield one more winning pair than that with $a_{\text{th}}^{\max} = +\infty$. The same phenomenon can be also found for the seller side.

In summary, the clearing prices are determined by

$$\begin{cases} A = B = v, \\ \qquad \text{if} \;\; a_{(i_g)} \leq v \leq b_{(j_g)} \text{ and } g < |\mathcal{P}_3|, & (15) \\ A = \max\{a_{(j_g)}, b_{\text{th}}^{\min}\}, \;\; B = \min\{b_{(i_g)}, a_{\text{th}}^{\max}\}, \\ \qquad \text{otherwise}, & (16) \end{cases}$$

Subroutine3 is enlightened by the TASC scheme [26]. However, unlike TASC, Subroutine3 exploits thresholds $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\max}$ to enhance the system efficiency.

To illustrate the idea of Subroutine3, we give two examples in Fig. 5, where the mapping relationships are the same with graph $\mathcal{G}_3$ in Fig. 2. In both subfigures, the buyers are sorted in non-increasing order according to their bid prices, i.e., $\{i_1, i_2, i_3, i_4, i_5\} = \{1, 2, 4, 5, 8\}$, and the sellers are sorted in non-decreasing order according to their ask prices, i.e., $\{j_1, j_2, j_3, j_4, j_5\} = \{1, 9, 3, 2, 8\}$. Assume that CSP sets $b_{\text{th}}^{\min} = 2.0$ and $a_{\text{th}}^{\max} = 4.5$. In Fig. 5 (a), we have $g = 5 = |\mathcal{P}_3|$, $b_{(i_5)} = b_{(8)} \geq a_{\text{th}}^{\max}$ and $a_{(j_5)} = a_{(8)} > b_{\text{th}}^{\min}$. Therefore, we obtain that $(\hat{x}, \hat{y}) = (5, 4)$, yielding the winning pairs $P_{1,3}$, $P_{2,1}$, $P_{4,9}$ and $P_{8,2}$. $P_{5,8}$ loses because seller 8 is at the right hand side of boundary $(5, 4)$. The clearing prices are determined by $B = a_{\text{th}}^{\max} = 4.5$ and $A = a_{(8)} = 3.3$. In Fig. 5(b), we have $g = 4 < |\mathcal{P}_3|$ and $a_{(i_4)} = a_{(2)} \leq v \leq b_{(j_4)} = b_{(5)}$, where $v = \frac{b_{(8)} + a_{(8)}}{2} = 4.3$. Therefore, we obtain the boundary $(\hat{x}, \hat{y}) = (4, 4)$, yielding the winning pairs $P_{1,3}$, $P_{2,1}$ and $P_{4,9}$, with the clearing prices $B = A = v = 4.3$.

After step two, the resource allocation and routing strategy $\boldsymbol{x}$ and $\boldsymbol{f}$ obtained from **P2** may result in resource wastage, because some candidate pairs are eliminated to ensure the

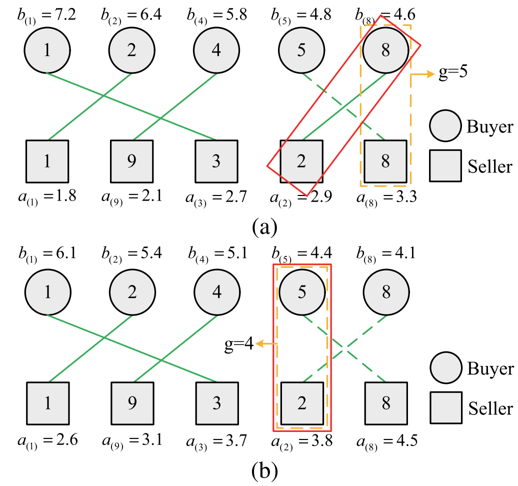


Fig. 5. Two examples of Subroutine3. In each subfigure, the dash rectangle marks the pair (i_g, j_g) with the maximum g that $b_{(i_g)} \geq a_{(j_g)}$, the solid rectangle marks the boundary $(\hat{x}, \hat{y})$, the solid lines denote the winning pairs, and the dash lines denote the losing pairs.

desired economic properties. Therefore, we can withdraw the network resources allocated to the losing candidate pairs. Specifically, based on the solution $\boldsymbol{f}$ to Problem **P2** and the winning pair determination outcome, we set $f_l^{i,\kappa_{i,j}}$ to 0 for each transmission link l if candidate pair $P_{i,j}$ fails, yielding the final flow routing strategy $\boldsymbol{f}^*$. Then, based on the solution $\boldsymbol{x}$ to Problem **P2**, we minimize the network resource usage (e.g., spectrum allocation) for each transmission link l while still supporting $\boldsymbol{f}^*$ over link l, leading to the final resource allocation strategy $\boldsymbol{x}^*$.

VI. Case Study: Edge Computing Over Wireless Mesh Networks

Problem **P1** in Section IV is a general formulation that can be easily adapted to various scenarios. To show a concrete **P1**, let us study a wireless mesh network where relay nodes are wirelessly interconnected to deliver data streams from buyers' data sources to sellers' servers. We also allow a source node to directly communicate with an edge server in close proximity. We consider $\boldsymbol{x}$ as spectrum allocation variables and present the constraints on $\boldsymbol{x}$ below.

Let $\mathcal{W}$ be the set of spectrum bands in the system. Define $x_l^w \in \{0, 1\}$ as the spectrum allocation variable, where $x_l^w = 1$ if band $w \in \mathcal{W}$ is allocated to link l, and $x_l^w = 0$ otherwise. A node cannot transmit to or receive from multiple neighboring nodes on the same band, i.e.,

$$\sum_{\{l \in \boldsymbol{L}_m^t | w \in \mathcal{W}_{n_l^r}\}} x_l^w \leq 1, \quad \forall m \in \mathcal{N}, w \in \mathcal{W}_m, \tag{17}$$

$$\sum_{\{l \in \boldsymbol{L}_n^r | w \in \mathcal{W}_{n_l^t}\}} x_l^w \leq 1, \quad \forall n \in \mathcal{N}, w \in \mathcal{W}_n, \tag{18}$$

where $\boldsymbol{L}_n^t$ or $\boldsymbol{L}_n^r$ is the set of links whose transmitter or receiver is node n. n_l^t or n_l^r denotes the transmitter or receiver of link l. $\mathcal{W}_m$ is the set of bands available around node m. To avoid self interference, a node cannot use the same band for transmission and reception simultaneously, i.e.,

$$x_l^w + \sum_{\{l' \in \boldsymbol{L}_n^t | w \in \mathcal{W}_{n_{l'}^r}\}} x_{l'}^w \leq 1, \quad \forall n \in \mathcal{N}, l \in \boldsymbol{L}_n^r, w \in \mathcal{W}_n \cap \mathcal{W}_{n_l^t}. \tag{19}$$

In addition, the interference from other nodes should be mitigated. Specifically, if node $n \in \mathcal{N}$ is receiving data on band $w \in \mathcal{W}_m$, neighboring nodes that may interfere with node n on band w should not use this band, i.e.,

$$x_l^w + \sum_{\{l' \in \boldsymbol{L}_{m'}^t | w \in \mathcal{W}_{n_{l'}^r}\}} x_{l'}^w \leq 1, \quad \forall m \in \mathcal{N}, l \in \boldsymbol{L}_m^t, \\ w \in \mathcal{W}_m \cap \mathcal{W}_{n_l^r}, m' \in \{m' | n_l^r \in \Upsilon_{m'}, m' \neq m, w \in \mathcal{W}_{m'}\}, \tag{20}$$

where $\Upsilon_{m'}$ denotes the set of nodes within the interference range of node m'. The protocol interference model is adopted here [39]: given the transmit power, the interference range is derived in the way that the received power within the range exceeds a certain threshold, and the interference is assumed to be negligible out of this range.

Furthermore, given $\boldsymbol{x}$, the aggregated data flow over each link is feasible only when it does not exceed the link capacity, i.e.,

$$\sum_{i \in \mathcal{I}} \sum_{1 \leq k \leq K_i} f_l^{i,k} \leq \sum_{w \in \mathcal{W}_{n_l^t} \cap \mathcal{W}_{n_l^r}} x_l^w e_l B_w, \quad l \in \mathcal{L} \tag{21}$$

where B_w is the bandwidth of band w, and e_l is the spectral efficiency of link l, depending on the transmit power of the transmitter and path loss over link l.

By replacing (6) with (21), and (11) with (17)-(20) in Problem **P1**, we obtain a joint service routing and spectrum allocation problem. At step one of COMSA, adding the linear constraint (14) produces the concrete form of Problem **P2**, which is a mixed integer linear program (MILP). While MILP is generally NP-hard, the solution to MILP has been extensively studied in the literature. For example, one can exploit the branch and bound scheme to find the global optimum [40] or coarse-grained fixing algorithm to obtain the suboptimal solution [3], [41]. With the solution to Problem **P2**, we can then go through the procedure in Section V to obtain the auction outcome.

VII. Mechanism Analysis

In this section, we first analyze the computational complexity of COMSA, and then show that COMSA is individually rational, budget balanced, and truthful.

Theorem 1: COMSA has the complexity of $\mathcal{O}(\mathcal{T} + |\mathcal{I}||\mathcal{J}| + |\mathcal{I}|\log|\mathcal{I}| + |\mathcal{J}|\log|\mathcal{J}|)$*, where* $\mathcal{T}$ *is the complexity for solving the optimization problem* ***P2*** *at step one.*

Proof: COMSA consists of two steps. At step one, COMSA solves Problem **P2** and then determines the buyer-seller pair partitioning as done in Fig. 2. Since the complexity of solving **P2** depends on the specific problem formulation and solution approach, we denote it by $\mathcal{O}(\mathcal{T})$ for generality. In our simulations, we adapt the coarse-grained fixing algorithm in [3] to obtain the approximate solution to Problem **P2** with the configurations in Section VI, which solves a series of linear programs (LPs) and a small-scale MILP (please refer to Section VIII-A for details). Given that an LP can be solved with the complexity of $\mathcal{O}(n^3)$ [42], where n is the number of variables, the complexity of the coarse-grained fixing algorithm is $\mathcal{O}((|\mathcal{N}_s|(|\mathcal{J}| + |\mathcal{L}|) + |\mathcal{L}||\mathcal{W}|)^3|\mathcal{L}||\mathcal{W}| + 2^{|\mathcal{N}_s||\mathcal{J}|}(|\mathcal{L}||\mathcal{N}_s|)^3)$ (the proof is omitted). The second term in $\mathcal{O}(\cdot)$ follows from the fact that the MILP can be solved by the branch and bound approach, which, in the worst case, degenerates to exhaustive search on $|\mathcal{N}_s||\mathcal{J}|$ binary variables together with solving LPs. However, since the branch and bound approach prunes the search space by eliminating candidate solutions that will not contain the optimal solution, the average computation complexity is far less than this. In fact, the coarse-grained fixing algorithm can solve **P2** with practical network scale efficiently, as validated by simulations in Section VIII-B. After obtaining the solution to **P2**, the pair partitioning process takes $\mathcal{O}(\max\{|\mathcal{I}|, |\mathcal{J}|\})$, which involves finding buyers and sellers associated with multiple agents. Then, at step two, COMSA is composed of three subroutines. In Subroutine1, finding the minimum bid price at each tree takes at most $\mathcal{O}(|\mathcal{I}|)$, and comparing the asks prices with the minimum bid price at each tree also takes at most $\mathcal{O}(|\mathcal{I}|)$. The number of trees is at most $|\mathcal{J}|$. The analysis for subroutine2 is similar to subroutine1. Therefore, the complexity of both subroutine1 and subroutine2 are $\mathcal{O}(|\mathcal{I}||\mathcal{J}|)$. In subroutine3, sorting the buyers and sellers take $\mathcal{O}(|\mathcal{I}|\log|\mathcal{I}|)$ and $\mathcal{O}(|\mathcal{J}|\log|\mathcal{J}|)$, respectively, while searching for the boundary and determining the winning pairs take $\mathcal{O}(\min\{|\mathcal{I}|, |\mathcal{J}|\})$. Consequently, the complexity of COMSA is $\mathcal{O}(\mathcal{T} + \max\{|\mathcal{I}|, |\mathcal{J}|\} + |\mathcal{I}||\mathcal{J}| + |\mathcal{I}|\log|\mathcal{I}| + |\mathcal{J}|\log|\mathcal{J}| + \min\{|\mathcal{I}|, |\mathcal{J}|\}) = \mathcal{O}(\mathcal{T} + |\mathcal{I}||\mathcal{J}| + |\mathcal{I}|\log|\mathcal{I}| + |\mathcal{J}|\log|\mathcal{J}|)$. The proof is completed. As can be observed, step two of COMSA is computationally efficient, whereas the complexity of step one depends on the specific problem and solution approach. □

Theorem 2: COMSA is individually rational.

Proof: To prove the individual rationality of COMSA, we should demonstrate that Subroutine1, Subroutine2, and Subroutine3 are individually rational.

Subroutine1 determines the winning pairs and pricing for each tree of graph $\mathcal{G}_1$ independently. In tree $\mathcal{G}_1^j$, the clearing price for both buyers and sellers, denoted by P_1^j, is $P_j = \min\{b_{s_j,\kappa_{s_j,j}}, a_{\text{th}}^{\max}\}$. Since $b_{s_j,\kappa_{s_j,j}}$ is the lowest bid price in $\mathcal{G}_1^j$, any winning buyer in $\mathcal{G}_1^j$ is charged no more than this bid price. Meanwhile, if a seller in $\mathcal{G}_1^j$ asks more than $b_{s_j,\kappa_{s_j,j}}$ or $a_{\text{th}}^{\max}$, it will lose. Overall, a winning seller is paid no less than its ask price. Therefore, Subroutine1 is individually rational. The proof for Subroutine2 is similar.

In Subroutine3, the clearing prices for buyers, i.e., B, and for sellers, i.e., A, are determined by (15) or (16). According to the winner determination policy, the winning buyer should bid no less than B, and the winning sellers should ask no more than A. Therefore, Subroutine3 is individually rational, which completes the proof. □

Theorem 3: COMSA achieves budget balance.

Proof: Clearly, COMSA achieves budget balance if the three subroutines ensure budget balance, respectively.

In Subroutine1 and Subroutine2, the clearing price are the same for both the buyer and the seller in any winning buyer-seller pair. Thus, the profit of the auctioneer equals zero.

In Subroutine3, if the payment is determined by (15), the clearing price are the same for winning buyers and sellers. Thus, the profit of the auctioneer equals zero. If the payment is determined by (16), the profit of the auctioneer is $U^A = \sum_{P_{i,j} \in \mathcal{S}_3} (\min\{b_{(i_g)}, a_{\text{th}}^{\max}\} - \max\{a_{(j_g)}, b_{\text{th}}^{\min}\}) \geq 0$, where S_3 is the set of winning pairs obtained from Subroutine3. Since $b_{(i_g)} \geq a_{(j_g)}$, $a_{\text{th}}^{\max} > b_{\text{th}}^{\min}$, $a_{\text{th}}^{\max} \geq a_{(j_g)}$ and $b_{\text{th}}^{\min} \leq b_{(i_g)}$, the auctioneer gains non-negative profit, which completes the proof. □

To show the truthfulness of COMSA, we should prove a series of lemmas.

Lemma 1: COMSA is truthful if Subroutine1, Subroutine2 and Subroutine3 are truthful, respectively.

Proof: The first step of COMSA is independent of bid/ask prices in the sense that a buyer/seller cannot gain control of the solution to the bid-independent optimization problem **P2** and the pair partitioning via bid manipulation. In other words, both buyers and sellers cannot control which subroutine their bids/asks enter. After the ungameable step one, the auction result (winning or not and pricing) of a bid/ask hinges on the corresponding subroutine while being independent of other subroutines. As a result, COMSA is truthful if Subroutine1, Subroutine2 and Subroutine3 are truthful, respectively. The proof is completed. □

Lemma 2: In Subroutine1, the auction result of pair $P_{i,j}$ is independent of buyer i's other bid prices and seller j's other ask prices, i.e., the bid price vector $\boldsymbol{B}_i^{-(i,j)} = \boldsymbol{B}_i \backslash \{b_{i,\kappa_{i,j}}\}$ and the ask price vector $\boldsymbol{A}_j^{-(i,j)} = \boldsymbol{A}_j \backslash \{a_{i,j,\kappa_{i,j}}\}$.

Proof: As aforementioned, the first step is independent of bids/asks. At the second step, suppose that pair $P_{i,j}$ falls into set $\mathcal{P}_1$ (hence in tree $\mathcal{G}_1^j$ of graph $\mathcal{G}_1$). On the one hand, the only bid price of buyer i in tree $\mathcal{G}_1^j$ is $b_{i,\kappa_{i,j}}$. On the other hand, although seller j may have multiple ask prices in $\mathcal{G}_1^j$, the auction outcome of $P_{i,j}$ only depends on $a_{i,j,\kappa_{i,j}}$ and the bid prices in $\mathcal{G}_1^j$. Therefore, it is impossible for buyer i and seller j to manipulate their bid/ask prices other than $b_{i,\kappa_{i,j}}$ and $a_{i,j,\kappa_{i,j}}$ to affect the auction outcome of pair $P_{i,j}$. The proof is completed. □

Lemma 3: Subroutine1 is truthful for buyers and sellers.

Proof: We first prove that Subroutine1 is truthful for buyers. To show the truthfulness, it is sufficient to demonstrate that buyer $i \in \mathcal{I}$ can maximize its utility in terms of any of its services, i.e., $U_{i,k}^b = y_{i,k}^b(v_{i,k} - \overline{b}_{i,k})$, by bidding truthfully. Without loss of generality, let us consider $k = \kappa_{i,j}$. With Lemma 2, we only need to investigate the impact of $b_{i,\kappa_{i,j}}$ on $U_{i,\kappa_{i,j}}^b$. Let $\hat{U}_{i,\kappa_{i,j}}^b$ denote the value of $U_{i,\kappa_{i,j}}^b$ when buyer i revealing its true valuation $v_{i,\kappa_{i,j}}$, and let $\tilde{U}_{i,\kappa_{i,j}}^b$ denote the value of $U_{i,\kappa_{i,j}}^b$ when it bids untruthfully. We examine all the possible cases listed in Table II to show that $\hat{U}_{i,\kappa_{i,j}}^b \geq \tilde{U}_{i,\kappa_{i,j}}^b$.

- Case 1: Buyer i loses its $\kappa_{i,j}$-th service regardless of its behavior, leading to $\hat{U}_{i,\kappa_{i,j}}^b = \tilde{U}_{i,\kappa_{i,j}}^b = 0$.
- Case 2: Due to the individual rationality that has proven in Theorem 2, we have $\hat{U}_{i,\kappa_{i,j}}^b \geq \tilde{U}_{i,\kappa_{i,j}}^b = 0$.

TABLE II

LOGIC OF CASE ANALYSIS. √ REPRESENTS THAT THE AGENT WINS, X REPRESENTS THAT THE AGENT LOSES

Case	1	2	3	4
Agent bids/asks truthfully	X	√	X	√
Agent bids/asks untruthfully	X	X	√	√

- Case 3: There are two possible reasons that buyer i loses when bidding truthfully: 1) $a_{i,j,\kappa_{i,j}}$ is higher than the lowest bid price in $\mathcal{G}_1^j$, i.e., $a_{i,j,\kappa_{i,j}} > b_{s_j,\kappa_{s_j,j}}$; 2) $v_{i,\kappa_{i,j}}$ equals the lowest bid price in $\mathcal{E}_j^1$, and $v_{i,\kappa_{i,j}} < a_{\text{th}}^{\max}$. In scenario 1), it is impossible for buyer i to win its $\kappa_{i,j}$-th service by manipulating its bid price. In scenario 2), buyer i has to bid higher than the second lowest bid price in $\mathcal{G}_1^j$ or $a_{\text{th}}^{\max}$ to win. In such a case, the clearing price equals $\min\{b', a_{\text{th}}^{\max}\}$, where b' denotes the lowest bid price in $\mathcal{G}_1^j$ other than $b_{i,\kappa_{i,j}}$, satisfying $b' > v_{i,\kappa_{i,j}}$. It therefore follows that $\tilde{U}_{i,\kappa_{i,j}}^b = v_{i,\kappa_{i,j}} - \min\{b', a_{\text{th}}^{\max}\} \leq 0 \leq \hat{U}_{i,k_{i,j}}^b$.
- Case 4: Recall that b' denote the lowest bid price in $\mathcal{G}_1^j$ except the bid of buyer i. If $b' < a_{\text{th}}^{\max}$, buyer i should bid no less than b' to win, and is charged b'. If $b' \geq a_{\text{th}}^{\max}$, buyer i should bid no less than $a_{\text{th}}^{\max}$, and is charged $a_{\text{th}}^{\max}$. The clearing price does not change no matter how buyer i bids. Thus, we obtain $\hat{U}_{i,\kappa_{i,j}}^b = \tilde{U}_{i,\kappa_{i,j}}^b$.

From the above, we show that $\hat{U}_{i,\kappa_{i,j}}^b \geq \tilde{U}_{i,\kappa_{i,j}}^b$ always holds. Therefore, Subroutine1 is truthful for buyers. Since the reasoning for the truthfulness of sellers is symmetric to that of buyers, we omit the detailed proof for sellers. The proof is completed. □

Lemma 4: In Subroutine2, the auction result of pair $P_{i,j}$ is independent of buyer i's other bid prices and seller j's other ask prices, i.e., the bid price vector $\boldsymbol{B}_i^{-(i,j)} = \boldsymbol{B}_i \backslash \{b_{i,\kappa_{i,j}}\}$ and the ask price vector $\boldsymbol{A}_j^{-(i,j)} = \boldsymbol{A}_j \backslash \{a_{i,j,\kappa_{i,j}}\}$.

Proof: The proof is similar to that for Lemma 2, and hence is omitted. □

Lemma 5: Subroutine2 is truthful for buyers and sellers.

Proof: With Lemma 4, we can prove Lemma 5 following a procedure similar to that for Lemma 3. The detail is omitted. □

Lemma 6: In Subroutine3, the auction result of pair $P_{i,j}$ (winning or not and pricing) is independent of buyer i's other bid prices and seller j's other ask prices, the bid price vector $\boldsymbol{B}_i^{-(i,j)} = \boldsymbol{B}_i \backslash \{b_{i,\kappa_{i,j}}\}$ and $\boldsymbol{A}_j^{-(i,j)} = \boldsymbol{A}_j \backslash \{a_{i,j,\kappa_{i,j}}\}$.

Proof: In graph $\mathcal{G}_3$, the only bid price from buyer i is $b_{i,\kappa_{i,j}}$, and the only ask price from seller j is $a_{i,j,\kappa_{i,j}}$. Thus, the auction outcome of pair $P_{i,j}$ is obviously independent of buyer i's and seller j's other prices, i.e., $\boldsymbol{B}_i^{-(i,j)} = \boldsymbol{B}_i \backslash \{b_{i,k_{i,j}}\}$ and $\boldsymbol{A}_j^{-(i,j)} = \boldsymbol{A}_j \backslash \{a_{i,j,k_{i,j}}\}$. The proof is completed. □

Lemma 7: In Subroutine3, if buyer i can win its $\kappa_{i,j}$-th service by bidding both $v_{i,\kappa_{i,j}}$ and $\tilde{b}_{i,\kappa_{i,j}}$, where $\tilde{b}_{i,\kappa_{i,j}} \neq v_{i,\kappa_{i,j}}$, it is charged the same price.

Proof: Let g denote the maximum index that $b_{(i_g)} \geq a_{(j_g)}$ when buyer i bidding $v_{i,k_{i,j}}$. According to (15) and (16), there are three possible values for buyer i's clearing price when it bids $v_{i,k_{i,j}}$: 1) $a_{\text{th}}^{\max}$; 2) $b_{(i_g)}$; and 3) v. In case 1), we know that all the bid prices of the buyers in the sorting $\mathbb{I}$ are higher than $a_{\text{th}}^{\max}$. As a result, buyer i has to bid $\tilde{b}_{i,k_{i,j}} > a_{\text{th}}^{\max}$ to win, in which case it is charged $a_{\text{th}}^{\max}$. In case 2), buyer i will be charged $b_{(i_g)}$ if bidding $\tilde{b}_{i,\kappa_{i,j}} \geq b_{(i_g)}$, and will lose otherwise. In case 3), buyer i will be charged v if bidding $\tilde{b}_{i,\kappa_{i,j}} \geq v$, and will lose otherwise. As a consequence, if buyer i wins, it will be charged the same price when bidding either $v_{i,k_{i,j}}$ or $\tilde{b}_{i,k_{i,j}}$. □

Lemma 8: In Subroutine3, if seller j can win the κ-th service of buyer i by both asking $c_{i,j,\kappa_{i,j}}$ and $\tilde{a}_{i,j,\kappa_{i,j}}$, where $\tilde{a}_{i,j,\kappa_{i,j}} \neq c_{i,j,\kappa_{i,j}}$, it is paid the same price.

Proof: The proof is similar to that for Lemma 7, and hence is omitted. □

Lemma 9: Subroutine3 is truthful for buyers and sellers.

Proof: We first show that Subroutine3 is truthful for buyers. Again, we examine all the possible cases listed in Table II to prove that $\hat{U}^b_{i,\kappa_{i,j}} \geq \tilde{U}^b_{i,\kappa_{i,j}}$ holds.

- Case 1: We have $\hat{U}^b_{i,\kappa_{i,j}} = \tilde{U}^b_{i,\kappa_{i,j}} = 0$.
- Case 2: Due to the individual rationality, we obtain $\hat{U}^b_{i,\kappa_{i,j}} \geq \tilde{U}^b_{i,\kappa_{i,j}} = 0$.
- Case 3: According to (15) and (16), there are three possible values for buyer i's clearing price when it bids $\tilde{b}_{i,\kappa_{i,j}}$: 1) $a_{\text{th}}^{\max}$; 2) $b_{(i_{\tilde{g}})}$, where $\tilde{g}$ is the maximum index that $b_{(i_{\tilde{g}})} \geq a_{(j_{\tilde{g}})}$ when buyer i bids $\tilde{b}_{i,\kappa_{i,j}}$; 3) v. Recall that buyer i loses by bidding $v_{i,\kappa_{i,j}}$. Therefore, subcase 1) occurs only if $\tilde{b}_{i,\kappa_{i,j}} \geq a_{\text{th}}^{\max} > v_{i,\kappa_{i,j}}$, which follows that $\tilde{U}^b_{i,\kappa_{i,j}} = v_{i,\kappa_{i,j}} - a_{\text{th}}^{\max} < 0 = \hat{U}^b_{i,\kappa_{i,j}}$. Subcase 2) happens only when $\tilde{b}_{i,\kappa_{i,j}} \geq b_{(i_{\tilde{g}})} \geq v_{i,\kappa_{i,j}}$, yielding $\tilde{U}^b_{i,\kappa_{i,j}} = v_{i,\kappa_{i,j}} - b_{(i_{\tilde{g}})} \leq 0 = \hat{U}^b_{i,\kappa_{i,j}}$. Subcase 3) occurs only when $\tilde{b}_{i,\kappa_{i,j}} \geq v \geq v_{i,\kappa_{i,j}}$, leading to $\tilde{U}^b_{i,\kappa_{i,j}} = v_{i,\kappa_{i,j}} - v \leq 0 = \hat{U}^b_{i,\kappa_{i,j}}$. For all the subcases, we obtain $\hat{U}^b_{i,\kappa_{i,j}} \geq \tilde{U}^b_{i,\kappa_{i,j}}$.
- Case 4: According to Lemma 7, the clearing prices are the same for both cases, leading to $\hat{U}^b_{i,\kappa_{i,j}} = \tilde{U}^b_{i,\kappa_{i,j}}$.

Then, we should prove that Subroutine3 is truthful for sellers. As shown in Subroutine3, sellers are symmetric to buyers in terms of both winner determination and pricing. Therefore, the proof for sellers' truthfulness is similar to that for buyers, and hence is omitted. The proof is completed. □

Using the above lemmas, we show the final result on COMSA's truthfulness.

Theorem 4: COMSA is truthful.

Proof: Lemma 1, Lemma 3, Lemma 5 and Lemma 9 together proves that COMSA is truthful. □

VIII. Performance Evaluation

In this section, we evaluate the system performance of COMSA, and study the economic and network impact on the MEC service provisioning.

A. Simulation Setup and Overview

We conduct the performance evaluation based on the wireless mesh network in Section VI. We consider a network where 4 relay nodes, J sellers (each owning one edge server), and I buyers (each having 2 source nodes) are deployed in a $1000 \times 1000m^2$ area. The transmission range and interference range of source nodes are set to 200m and 300m, while the transmission range and interference range of relay nodes are set to 500m and 600m. For illustrative purpose, all the nodes are assumed to have the same set of W spectrum bands, each having the identical bandwidth of 5MHz. We assume that the transmit power at source nodes and relay nodes are 0.6W and 5W, respectively. For each link l, we adopt the deterministic power propagation model [43] with path loss exponent 4, noise power spectral density $10^{-16}W/Hz$, and antenna-related parameter 1 to calculate the channel capacity e_l in (21). The computing capabilities of server nodes, namely, Θ_j, are uniformly generated in $[6, 14]$GHz. The memory spaces Φ_j at server nodes are uniformly drawn from $[8, 24]$GB. For each service request, the data rate requirement $r_{i,k}$, computing power requirement $\theta_{i,k}$ and memory requirement $\phi_{i,k}$ are drawn from the intervals $[1, 2]$Mbps, $[1, 4]$GHz, and $[1, 3]$GB, respectively [3], [14], [44]. To investigate the impact of bid/ask distributions, we assume that true valuations (and bid prices) for unit data rate are uniformly distributed in $[0.5, \beta]$, and the true costs (and ask prices) for unit data rate are uniformly distributed in $(0, 1]$, where β is set to 4 by default. We set weight $M_{i,k} = r_{i,k}$ for each service, and hence the objective of COMSA is to maximize the system service throughput. The following numerical results are averaged over 10 random network topologies.

TABLE III
The Processed Bits per Second Versus the Number of Bands W Obtained From Problem **P1** and Problem **P2**, With $I = 5$ (10 Requests) and $J = 4$

Number of bands	1	2	3	4	5
Optimal (**P1**)	3.23	4.93	5.90	6.59	7.13
Heuristic (**P1**)	2.61	3.96	5.02	5.54	6.72
Optimal (**P2**)	3.23	4.93	5.90	6.59	7.13
Heuristic (**P2**)	2.61	3.96	4.65	5.38	6.72

We adapt the coarse-grained fixing algorithm in [3], [41] to obtain an approximate solution to Problem **P2** with the configurations in Section VI. The main obstacle in solving **P2** optimally lies in the binary spectrum allocation variables x_l^w, because the number of network links can be very large. The core idea of this algorithm is to relax binary variable $d_{i,k}^j$ and x_l^w to $[0, 1]$, and solve the relaxed problems iteratively by fixing at least one x_l^w to 1 in each round. Specifically, in each round, for x_l^w's greater than α in the obtained solution, fix them to 1, where $\alpha > 0.5$. If none of x_l^w's are greater than α, choose the maximum x_l^w and fix it to 1. Meanwhile, fix a set of other variables x_l^w to 0 according to spectrum allocation constraints. Until all x_l^w are fixed to either 1 or 0, we solve Problem **P2** with continuous variables $f_l^{i,k}$ and binary variables $d_{i,k}^j$ at the last step. Although it still requires solving an MILP with binary variables $d_{i,k}^j$, the number of $d_{i,k}^j$ is not excessively large under a practical problem setting. More details can be found in [3]. We call this algorithm as "heuristic algorithm" hereafter, and set α to 0.85 in our simulations [41]. In Table III, we compare the heuristic algorithm with the optimal results (which can be obtained by an MILP solver, i.e., MATLAB) in small-sized networks, where we set $I = 5$

(10 requests) and $J = 4$, and vary the number of bands W. For both problems, we observe that the performance achieved by the heuristic algorithm is close to the optimal solution. Since solving **P2** optimally would demand an intolerable time when the problem scale goes up, we will adopt the heuristic algorithm to solve Problem **P2** in the following simulations. Furthermore, it can be seen that the optimal throughput of Problem **P2** is equal to that of Problem **P1**, implying that the additional constraint (14) in **P2** has a negligible impact on the system performance under the considered simulation setting.

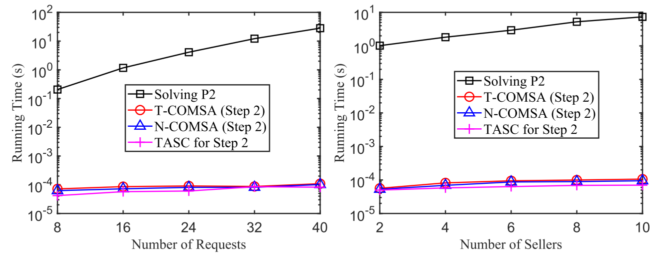
We will compare the system service throughput achieved by the following mechanisms:

- **Untruthful auction with truthful agents (UA-TA)** represents the system service throughput that can be achieved when the auction scheme does not ensure truthfulness but every agent behaves truthfully, which is impractical. As discussed in Section III-C, an untruthful auction could produce very poor outcome in reality, because agents have the incentives to strategize over others. Although UA-TA cannot be achieved in practice, it provides the upper bound on system efficiency to reveal the cost of achieving truthfulness in our proposed COMSA. Specifically, to formulate UA-TA, we add additional variables $\overline{b}_{i,k}$ and $\overline{a}_{i,j,k}$ and the following constraints to problem **P1**:

$$\overline{b}_{i,k} = b_{i,k}, \quad \overline{a}_{i,j,k} = a_{i,j,k}, \quad \forall i \in \mathcal{I}, 1 \leq k \leq K_i, j \in \mathcal{J}, \tag{22}$$

$$\sum_{i \in \mathcal{I}} \sum_{1 \leq k \leq K_i} \sum_{j \in \mathcal{J}} d_{i,k}^{j} (\overline{b}_{i,k} - \overline{a}_{i,j,k}) \geq 0. \tag{23}$$

  The obtained solution to this reformulated problem is UA-TA. Here, we directly set the clearing prices $\overline{b}_{i,k}$ and $\overline{a}_{i,j,k}$ to the bid and ask prices, as indicated by (22). Moreover, we use (23) to ensure the budget balance. It is clear that individual rationality and budget balance are satisfied according to (22) and (23). However, the truthfulness cannot be guaranteed, since a bidder can manipulate its bids/asks to control the clearing prices and affect the winning bid/ask list. By assuming that agents behave truthfully, we set bid price $b_{i,k}$ and ask price $a_{i,j,k}$ in (22) to the agents' true valuations and costs, respectively.
- **Threshold-Based COMSA (T-COMSA)** denotes the COMSA scheme with appropriate thresholds $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\min}$. In the simulations, we set $b_{\text{th}}^{\min} = 0.5$, and $a_{\text{th}}^{\max} = 1$ by default, unless specified in the figure.
- **No-Threshold-Based COMSA (N-COMSA)** represents the COMSA scheme with $b_{\text{th}}^{\min} = 0$ and $a_{\text{th}}^{\max} = \infty$.
- **TASC** is a truthful double auction scheme for cooperative communications [26], which consists of a bid-independent assignment step and a winner determination & pricing step. TASC can be adapted to our considered MEC scenario. However, TASC only supports one-to-one buyer-seller assignment. Thus, we add additional constraints to Problem **P2** to ensure that each buyer has at most one computing service request admitted and each seller serves at most one request. We take the assignment result obtained from this problem into the winner determination & pricing step of TASC to obtain the final auction outcome.



(a) The running time versus the number of buyers, with $J = 4$ and $W = 4$.

(b) The running time versus the number of sellers, with $I = 15$ (30 requests) and $W = 4$.

Fig. 6. The running time for different schemes.

Our following simulations have two major goals. First, we show the running time for COMSA. Second, we compare the system service throughput of COMSA with other benchmarks with varying network and economic factors. In particular, we examine the cost of ensuring truthfulness by showing the performance degradation of COMSA over UA-TA, and also demonstrate the superiority of COMSA by comparing it with TASC.

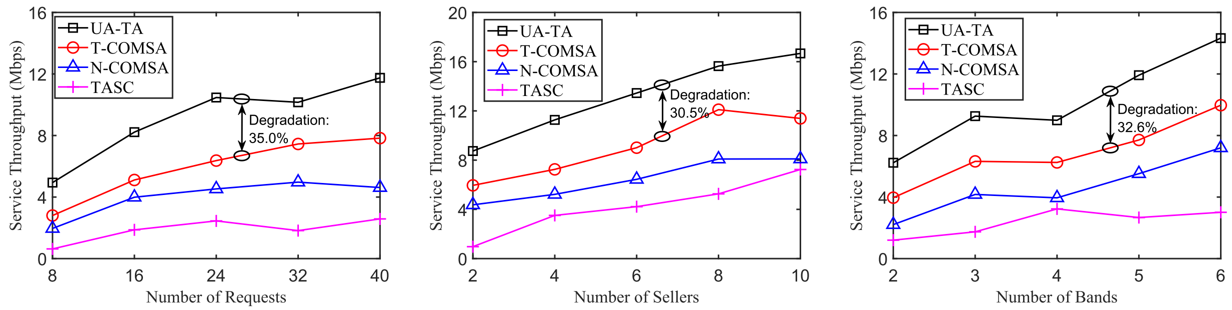
B. Running Time

Fig. 6 illustrates the running time for COMSA and TASC. Recalling that both COMSA and TASC (when adapted to our MEC systems) consist of two steps, we plot the running time for step one and step two separately in the figure. Step one is to solve Problem **P2**, where the running time depends on the specific network and solution approach. As mentioned earlier, in our simulations, we employ the coarse-grained fixing algorithm to solve Problem **P2**. By considering the wireless mesh networks in Section VI, we observe that the time taken to solve Problem **P2** dominates the running time. On the other hand, step two of both T-COMSA and N-COMSA are very efficient, which are comparable to that of TASC.

C. System Service Throughput

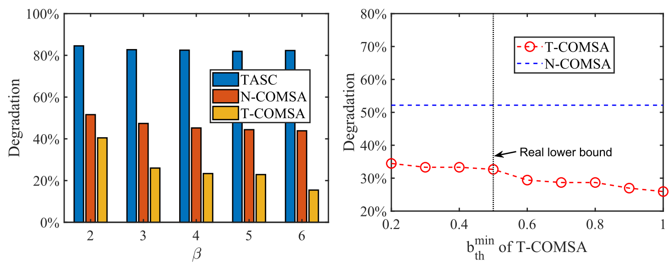
1) Impact of Network Factors: In Fig. 7, we compare the system service throughput for different schemes by varying the number of requests, sellers, and bands. From Fig. 7a to Fig. 7c, both T-COMSA and N-COMSA perform much better than TASC. This is because TASC only supports one-to-one pairing between buyers and sellers, which is not suitable for MEC systems where each edge server is generally capable of serving multiple users at the same time. Another key observation is that, T-COMSA achieves a noticeably better performance than N-COMSA. This is because T-COMSA can potentially sacrifice fewer buyer-seller pairs than N-COMSA by properly setting the thresholds $b_{\text{th}}^{\min}$ and $a_{\text{th}}^{\max}$.

Furthermore, Fig. 7 compares COMSA with UA-TA. The gap between COMSA and UA-TA is called "performance degradation", which is defined as $(S_0 - S_{\text{COMSA}})/S_0$, where S_0 and S_{COMSA} denote UA-TA and COMSA, respectively.



(a) The system service throughput versus the number of requests ($=2I$), with $J=4$ and $W=4$. (b) The system service throughput versus the number of sellers J, with $I=20$ (40 requests) and $W=4$. (c) The system service throughput versus the number of bands W, with $I=20$ (40 requests) and $J=4$.

Fig. 7. The system service throughput under different schemes with varied network factors.



(a) The performance degradation over UA-TA versus the upper bound of bid, i.e., β, with $I=20$ (40 requests), $J=4$, and $W=4$. (b) The performance degradation over UA-TA versus threshold b_{th}^{min}, with $a_{th}^{min}=1$, $I=20$ (40 requests), $J=4$, and $W=4$.

Fig. 8. The performance degradation over UA-TA under different bid distributions and bid threshold b_{th}^{min}.

Following the idea of "trade reduction" in double auction mechanism design [17], COMSA potentially eliminates some buyer-seller pairs and sets the clearing prices to the removed bids/asks or predefined thresholds to guarantee truthfulness. Thus, the degradation reflects the cost of guaranteeing truthfulness. Fortunately, the degradation of T-COMSA is $30.5\% - 35\%$ compared with UA-TA, which implies that guaranteeing truthfulness does not lead to significant performance loss.

2) Impact of Bid Distribution: Fig. 8a illustrates the performance degradation of COMSA over UA-TA in terms of the bid distributions. It is not surprising that the degradation declines with β. Since bid prices per unit data rate are drawn from the interval $[0.5, \beta]$, this phenomenon reveals that COMSA can achieve a better performance when the true valuations from buyers are generally higher than the true costs from sellers. Fig. 8b examines the impact of thresholds b_{th}^{min} on the performance of T-COMSA, where N-COMSA is used as the performance benchmark. Recalling that the bid prices per unit data rate are drawn from the interval $[0.5, 4]$. On the one hand, T-COMSA with b_{th}^{min} directly rules out the bids lower than b_{th}^{min}, implying that some bids will be directly eliminated if b_{th}^{min} is set to be larger than 0.5. On the other hand, T-COMSA may sacrifice less bids/asks than N-COMSA in subroutine1, subroutine2, and subroutine3. We can observe that T-COMSA always outperforms N-COMSA when b_{th}^{min} varies from 0 to 1. One important observation is that the minimal degradation is achieved at the point not necessarily equal to the real lower bound, i.e., 0.5, of the bid distribution. In other words, eliminating some extreme prices by setting a stricter threshold than the real bound might enhance the system performance, because a very low bid not only has little chance to win, but also can make other pairs in the same tree hard to win, according to the rule of Subroutine1. Overall, the adoption of appropriate thresholds is crucial to the performance of COMSA.

IX. Conclusion

Most existing double auction mechanisms in the wireless community focus on either the computing resources or communications resources only. Even if very few of them do take both types of resources into consideration in the auction design, they at most treat the auction design as joint resource optimization problems where the bidders therein still need to select and bid for resources. Yet, what end users care the most is whether their services with the desired level of quality of service can be provided and they do not really care how much resources they are allocated. In response to this observation, in this paper, we proposed COMSA, a general "service-oriented" double auction mechanism to tackle the joint problem of incentive design and service provisioning for edge computing. Specifically, COMSA not only sets up a double-sided auction in which service sellers can gain satisfying rewards from service buyers to compensate their costs, but also allocates the limited computing and network resources to support the winning computing service requests with proper E2E QoS guarantees. We have proved that COMSA is truthful, individually rational, and budget balanced. Simulations demonstrate that COMSA can achieve the desirable economic properties while maintaining satisfactory system service throughput.

In addition to service throughput, another important metric in auction design is social welfare. In COMSA, the bid-independent optimization problem at step one aims at maximizing the weighted sum of accepted services without taking bids/asks into account. Consequently, the requests with high valuations may be rejected by step one of COMSA, thus resulting in social welfare degradation. Thus, how to design an E2E service auction for MEC with improved social welfare is a challenging yet interesting research problem.

References

[1] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 1st Quart., 2018.

[2] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1584–1607, Aug. 2019.

[3] H. Ding, Y. Guo, X. Li, and Y. Fang, "Beef up the edge: Spectrum-aware placement of edge computing services for the Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 18, no. 12, pp. 2783–2795, Dec. 2019.

[4] S. Liu, J. Tang, Z. Zhang, and J.-L. Gaudiot, "Computer architectures for autonomous driving," *Computer*, vol. 50, no. 8, pp. 18–25, Aug. 2017.

[5] H. Ding, C. Zhang, Y. Cai, and Y. Fang, "Smart cities on wheels: A newly emerging vehicular cognitive capability harvesting network for data transportation," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 160–169, Apr. 2018.

[6] H. Ding, X. Li, Y. Cai, B. Lorenzo, and Y. Fang, "Intelligent data transportation in smart cities: A spectrum-aware approach," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2598–2611, Dec. 2018.

[7] U. Habiba and E. Hossain, "Auction mechanisms for virtualization in 5G cellular networks: Basics, trends, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2264–2293, 3rd Quart., 2018.

[8] 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 Trans. Services Comput.*, vol. 9, no. 6, pp. 895–909, Nov./Dec. 2016.

[9] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4692–4701, Oct. 2018.

[10] X. Chen, Y. Deng, G. Zhu, D. Wang, and Y. Fang, "From resource auction to service auction: An auction paradigm shift in wireless networks," *IEEE Wireless Commun.*, early access, May 11, 2022, doi: 10.1109/MWC.005.2100627.

[11] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[12] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[13] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[14] R. Yu, G. Xue, and X. Zhang, "Application provisioning in FOG computing-enabled Internet-of-Things: A network perspective," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 783–791.

[15] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1440–1452, May 2016.

[16] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghrayeb, "Optimized provisioning of edge computing resources with heterogeneous workload in IoT networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 459–474, Jun. 2019.

[17] R. P. McAfee, "A dominant strategy double auction," *J. Econ. Theory*, vol. 56, no. 2, pp. 434–450, Apr. 1992.

[18] P. Huang, A. Scheller-Wolf, and K. Sycara, "Design of a multi-unit double auction E-market," *Comput. Intell.*, vol. 18, no. 4, pp. 596–617, Nov. 2002.

[19] M. Babaioff and W. E. Walsh, "Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation," *Decis. Support Syst.*, vol. 39, no. 1, pp. 123–149, Mar. 2005.

[20] L. Y. Chu and Z.-J.-M. Shen, "Truthful double auction mechanisms," *Oper. Res.*, vol. 56, no. 1, pp. 102–120, Feb. 2008.

[21] L. Y. Chu, "Truthful bundle/multiunit double auctions," *Manage. Sci.*, vol. 55, no. 7, pp. 1184–1198, Jul. 2009.

[22] P. Dütting, I. Talgam-Cohen, and T. Roughgarden, "Modularity and greed in double auctions," *Games Econ. Behav.*, vol. 105, pp. 59–83, Sep. 2017.

[23] X. Zhou and H. Zheng, "TRUST: A general framework for truthful double spectrum auctions," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 999–1007.

[24] X. Feng, Y. Chen, J. Zhang, Q. Zhang, and B. Li, "TAHES: A truthful double auction mechanism for heterogeneous spectrums," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 4038–4047, Nov. 2012.

[25] Y. Chen, J. Zhang, K. Wu, and Q. Zhang, "TAMES: A truthful double auction for multi-demand heterogeneous spectrums," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 3012–3024, Nov. 2014.

[26] D. Yang, X. Fang, and G. Xue, "Truthful auction for cooperative communications," in *Proc. 12th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, May 2011, pp. 1–10.

[27] P. Li, S. Guo, and I. Stojmenovic, "A truthful double auction for device-to-device communications in cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 71–81, Jan. 2016.

[28] D. Zhang *et al.*, "Near-optimal and truthful online auction for computation offloading in green edge-computing systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 880–893, Apr. 2020.

[29] X. Wang, X. Chen, and W. Wu, "Towards truthful auction mechanisms for task assignment in mobile device clouds," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.

[30] G. Gao, M. Xiao, J. Wu, H. Huang, S. Wang, and G. Chen, "Auction-based VM allocation for deadline-sensitive tasks in distributed edge cloud," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1702–1716, Nov. 2021.

[31] 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. 13455–13464, Jul. 2017.

[32] L. Lu, J. Yu, Y. Zhu, and M. Li, "A double auction mechanism to bridge users' task requirements and providers' resources in two-sided cloud markets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 4, pp. 720–733, Apr. 2018.

[33] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, "TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things," *IEEE Trans. Mobile Comput.*, early access, Mar. 8, 2021, doi: 10.1109/TMC.2021.3064314.

[34] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju, "DeCloud: Truthful decentralized double auction for edge clouds," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 2157–2167.

[35] J. Lin, L. Huang, H. Zhang, X. Yang, and P. Zhao, "A novel latency-guaranteed based resource double auction for market-oriented edge computing," *Comput. Netw.*, vol. 189, Apr. 2021, Art. no. 107873.

[36] "Understanding IP surveillance camera bandwidth," Fortinet, Sunnyvale, CA, USA, White Paper 740347-0-0-EN, May 2017. [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/white-papers/wp-ip-surveillance-camera.pdf

[37] X. Li, H. Ding, M. Pan, Y. Sun, and Y. Fang, "Users first: Service-oriented spectrum auction with a two-tier framework support," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 2999–3013, Nov. 2016.

[38] R. Madan, S. Cui, S. Lall, and A. Goldsmith, "Cross-layer design for lifetime maximization in interference-limited wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3142–3152, Nov. 2006.

[39] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless Netw.*, vol. 11, no. 4, pp. 471–487, Jul. 2005.

[40] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*, vol. 55. Hoboken, NJ, USA: Wiley, 1999.

[41] Y. T. Hou, Y. Shi, and H. D. Sherali, "Spectrum sharing for multi-hop networking with cognitive radios," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 146–155, Jan. 2008.

[42] Y. Ye, "An $O(n^3L)$ potential reduction algorithm for linear programming," *Math. Program.*, vol. 50, no. 1, pp. 239–258, Mar. 1991.

[43] H. Yue, M. Pan, Y. Fang, and S. Glisic, "Spectrum and energy efficient relay station placement in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 5, pp. 883–893, May 2013.

[44] H. Moens, B. Hanssens, B. Dhoedt, and F. D. Turck, "Hierarchical network-aware placement of service oriented applications in clouds," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–8.