Edge AI Inference as a Service via Dynamic Resources from Repeated Auctions

Mingtao Ji, Hehan Zhao, Lei Jiao, Sheng Zhang, Xin Li, Zhuzhong Qian, Baoliu Ye

Abstract—To enable edge AI providers to recruit edge devices and use them to deploy AI models and provision inference services, we conduct a comprehensive mathematical and algorithmic study on a novel incentive and optimization mechanism based on repeated auctions. We first model and formulate a time-cumulative social cost optimization problem to capture the challenges of the trade-off between cost and accuracy, the dependency between adjacent auctions, and the need of achieving desired economic properties. Then, to solve this intractable non-linear integer program in an online manner, we design a set of polynomial-time algorithms that work together. Our approach dynamically chooses and switches winning bids under careful control, incorporates online learning to overcome posterior inference accuracy and workload queue dynamics, and leverages randomization to strategically convert fractional decisions of model placement and query dispatch into integers. We also allocate payments to meet the necessary and sufficient conditions for the desired economic properties. Further, we rigorously prove the constant competitive ratio, the sub-linear regret and fit, and the truthfulness and individual rationality for our proposed approach. Finally, through extensive experiments using real devices, AI models, and data traces, we have validated the substantial advantages of our proposed approach compared to the baselines and the state-of-the-art methods.

Index Terms—Edge AI, Inference, Edge Computing, Online Optimization, Online Learning, Auction

1 Introduction

Edge AI inference [1] entails deploying machine learning models upon devices at the network edge closer to the end users, substantially reducing response time and bandwidth consumption and safeguarding data privacy [2]. As a service, edge AI inference is adopted in a range of applications, such as virtual reality [3], healthcare [4], and autonomous vehicles [5]. For example, in virtual reality, the inference service receives the end users' inference requests such as gestures and voices and provides inference results promptly.

Unfortunately, many edge AI services do not actually have sufficient and satisfying edge devices for deploying edge AI inference. Although today's cloud computing vendors also provide "pay-as-you-go" edge computing services [6], they could still be costly, especially for small-scale edge AI businesses. For instance, Amazon CloudFront has the computation price 6 times higher than its EC2 [7]. Besides, such existing edge infrastructures, distributed though, often reside at pre-specified fixed locations, and may still be unable to sufficiently reach the target group of end users in particular geographic locations at a finer granularity. More importantly, current commercial edge provisioning focuses on edge *servers* only; yet, an edge device does not actually have to be an edge server, and it can be a smart router [8], a home NAS [9], and a personal computer, e.g., the "AI PC" [10]. Ideally, it could be better if any of such edge devices at any location could be potentially and appropriately employed and released as needed by the edge AI service.

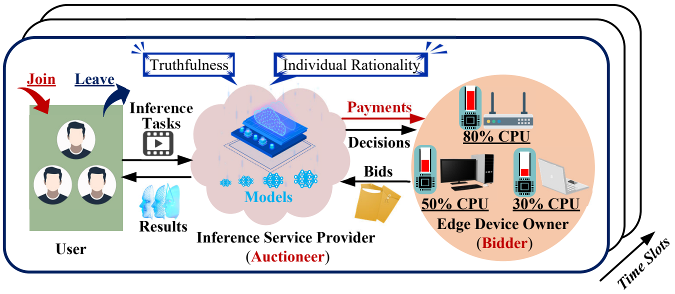


Fig. 1: Architecture of Auction-Based Edge AI Inference

Moving toward this direction, what is needed here is an incentive mechanism to incentivize the edge device owners to contribute their edge devices or idle resources on such edge devices to the edge AI service. *Auction* can serve as such an incentive mechanism, as shown in Fig. 1. The edge AI inference service can act as the auctioneer and conduct auctions to procure edge resources as bids from the edge devices that can act as the bidders. Auction has many advantages indeed, including reducing the chance of mispricing, matching the demand with the supply better, and capturing the market dynamics in real time [11], [12].

However, it is non-trivial to design auction-based mechanisms to continuously operate and orchestrate the edge AI inference service with desired service quality at minimum cost upon dynamically-recruited edge resources, while keeping the edge devices incentivized to contribute such resources. We identify three unique and fundamental challenges as follows, also highlighted in Fig. 2.

Cost vs. Accuracy under System Dynamics: Given the often limited edge capacity, we need to carefully choose between high-accuracy models which can consume excessive edge resources and low-accuracy models which can

- *M. Ji, H. Zhao, S. Zhang, Z. Qian, and B. Ye are with the State Key Laboratory for Novel Software Technology, School of Computer Science, Nanjing University, Nanjing 210023, China (E-mail: jmt@smail.nju.edu.cn, hehan.zhao@mail.utoronto.ca, {sheng, qzz, yebl}@nju.edu.cn).*
- *L. Jiao is with the Center for Cyber Security and Privacy, University of Oregon, Eugene, OR 97403, USA (E-mail: ljiao2@uoregon.edu).*
- *X. Li is with the Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China (E-mail: lics@nuaa.edu.cn).*

Corresponding authors: Zhuzhong Qian; Lei Jiao.

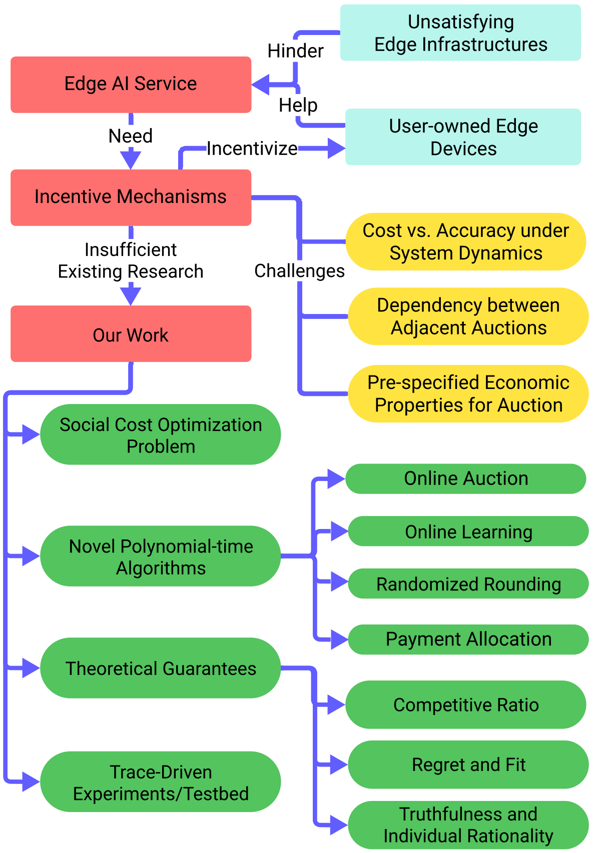

Unsatisfying Edge Infrastructures
Hinder
Edge AI Service
Help
User-owned Edge Devices
Need
Incentivize
Incentive Mechanisms
Cost vs. Accuracy under System Dynamics
Insufficient Existing Research
Challenges
Dependency between Adjacent Auctions
Our Work
Pre-specified Economic Properties for Auction
Social Cost Optimization Problem
Online Auction
Online Learning
Novel Polynomial-time Algorithms
Randomized Rounding
Payment Allocation
Theoretical Guarantees
Competitive Ratio
Regret and Fit
Trace-Driven Experiments/Testbed
Truthfulness and Individual Rationality

Fig. 2: Structure of the Introduction Section

save resources but have less satisfying service quality. What complicates this is that the inference accuracy is posterior—we only observe the accuracy (or the error rate) of a model upon users' inference queries *after* we actually deploy and use this model to serve those queries. When making model deployment decisions, as the actual queries have not been served yet, we have no knowledge about the model's performance on those queries. The difficulty further escalates because meanwhile we need to (i) dispatch users' queries to different edge devices under time-varying network conditions at minimum communication cost, and (ii) maintain and eventually clear the query queue on each edge device given models' diverse execution speeds on different devices.

Dependency between Adjacent Auctions: The auctions are not a one-time transaction, but are often repetitive as we continuously recruit edge resources; yet, each auction cannot just work independently. An edge device that was a winning bid in the last auction but does not win in the current auction may revoke its resources, shut down, or leave the system; if it joins again in the next auction and is selected to win, then it can incur "switching cost", because the inference service needs to re-authorize the device, re-initialize its execution environment, and re-deploy the model(s), causing leading time and effort. To mitigate such switching cost, ideally, we should choose a winning bid in one auction by cautiously expecting whether this bid could also win in the next auction; yet, this is pretty hard if ever possible, because the next auction has not occurred yet and the inputs then could all change as it happens [1], [13].

Economic Properties of Each Auction: Even for a single auction in our scenario, it is still non-trivial to design and solve. Choosing winning bids requires to not only balance cost and accuracy but also attain the desired economic properties such as truthfulness and individual rationality [11], [14]. Truthfulness ensures that every bid maximizes its utility by bidding the price that reflects the bidder's true valuation (i.e., no motivation to lie about the bidding price), and individual rationality ensures that every bid always has non-negative utility regardless of the auction outcome (i.e., voluntary auction participation). The well-known Vickrey-Clarke-Groves (VCG) mechanisms [15] can achieve such economic properties. However, original VCG requires to exactly solve the underlying optimization problem, but our problem, due to the complexities and integer decisions, is intractable as will be shown; fractional VCG [16] is also inapplicable, due to the existence of the extra terms of the edge AI inference overhead and accuracy in each auction.

Existing studies fall short to effectively tackle the aforementioned challenges. Many works [1], [13], [17]–[19] on edge AI inference systems often assume abundant resources without dynamically adding or removing resources through economic means, unable to meet the needs of our scenario. Other works [11], [12], [20]–[23] on incentive mechanisms at the network edge typically do not consider AI inference, or ignore the unique features and challenges of edge AI inference as we highlight above, thus inapplicable to our scenario. See Section 6 for detailed discussions.

In this paper, we overcome all the aforementioned challenges via a mathematical and algorithmic study with solid theoretical analysis. We make multiple contributions:

First, we model and formulate a time-cumulative social cost optimization problem, which minimizes the sum of the edge AI service's cost, the edge devices' cost, and the inference error rate, subject to resource capacity and inference query queue dynamics. The control decisions include winning-bid selection, model placement, query dispatch, and payments to bids in each auction at each corresponding time slot. This problem is a non-linear integer program, unsurprisingly NP-hard, but is general enough with almost zero assumption on the input dynamics and heterogeneities.

Second, to enable the edge AI service to solve this problem to provision the inference service while procuring the resources, we propose four novel polynomial-time online algorithms that can work jointly. We decouple our original problem into two sub-problems based on the bid-switching cost. We design our overall online optimization algorithm to dynamically balance the switching cost and all the rest of the cost terms by only triggering a new switch operation when the accumulated other costs exceed a pre-specified parameter times the switching cost of the last switch operation. To that end, we invoke our second algorithm, which is based on online learning, to address the posterior inference accuracy and the query queue state transitions for each auction. During this process, we also invoke our third algorithm which adopts randomization to convert all our fractional control decisions into integers without violating any constraints at each time slot. Via our fractional and integral bid-selection decisions, our fourth algorithm calculates payments in each auction using the real bidding prices and the conceived alternative bidding prices of the winning bids.

Third, we rigorously prove the multiple worst-case performance guarantees associated with our algorithms. Overall, our combined algorithmic approach achieves a constant competitive ratio, i.e., the social cost incurred by our online approach (which observes the inputs gradually) is upper-bounded by this constant times the social cost in the offline optimal situation (which observes all inputs in hindsight at once). For our online learning component that addresses all the non-switching costs, we achieve sub-linear regret and fit, i.e., as time goes, the time-averaged difference between those costs incurred by our approach and those incurred in the series of one-shot optimums vanishes, and the time-averaged violation of the long-term constraints also vanishes. Our payment allocation component meets the sufficient and necessary conditions for truthfulness and individual rationality in randomized auctions, and thus achieves these economic properties simultaneously.

Fourth, we build a testbed with real hardware, adopt real AI models, and compare our approach to a variety of other methods driven by real traces (e.g., edge device locations, workload variations) for various performance metrics. We briefly summarize our observations: (i) Our approach saves social cost in real time, reducing social cost by 55%∼70% cumulatively compared to baselines and by 30%∼49% cumulatively compared to state-of-the-arts; (ii) Our approach scales well with inputs and consistently outperforms others; (iii) Our online learning component has the lowest regret, achieving sub-linear growth in both regret and fit; (iv) Our approach achieves truthfulness and individual rationality in practice; (v) Our algorithms run fast and finish in tens of milliseconds, except the payment allocation which could consume seconds, acceptable per 15-minute-long time slot.

The rest of this paper is structured as follows. Section 2 models and formulates the social cost optimization problem. Section 3 proposes and designs our algorithms to solve this problem online. Section 4 analyzes and proves the theoretical performance guarantees of our algorithms. Section 5 validates the practical performance of our algorithms via testbed-based experiments. Section 6 summarizes the existing research. Section 7 discusses some related issues in this work. Section 8 concludes.

TABLE 1: Notations

Input	Description
$\mathcal{N}$	Set of edge devices
$\mathcal{M}$	Set of AI models
$\mathcal{T}$	Set of time slots
$\mathcal{B}_{n,t}$	Bid submitted by edge device n at time slot t
$b_{n,t}$	Bidding price of bid n at t
l_n	Switching cost for edge device n
c_n	Computing resource capacity of edge device n
r_m	Computing resource demand of model m
q_n	Query queue capacity of edge device n
$p_{n,m}$	# of queries processed by model m on edge device n per time slot
$e_{n,m,t}$	Comm. cost for sending model m to edge device n at t
$d_{n,t}$	Comm. cost for sending one query to edge device n at t
$\delta_{n,m,t}$	Whether to send model m to edge device n at t
$a_{n,m,t}$	Error rate of model m on edge device n at t
τ_t	# of queries submitted by end users at t
Decision	**Description**
$x_{n,t}$	Whether edge device n wins in the auction at t
$y_{n,m,t}$	Whether model m is deployed on edge device n at t
$z_{n,m,t}$	# of queries sent to model m on edge device n at t
$w_{n,t}$	Payment made to edge device n at t

2 Modeling and Formulation

2.1 Scenario Feasibility

Today's user-owned edge hardware already supports the deployment of edge AI inference and also has the idle capacity for it. These edge devices just need to be incentivized to participate in edge AI inference, which motivates our study.

- Abundant Idle Resources at Edge: The number of the edge devices in reality, such as smart routers and home NAS systems [9], has been growing steadily in recent years. According to the Global IoT Market Forecast report [24], the number of IoT devices is expected to reach 40 billion by 2030. Yet, a significant portion of such devices remain idle [25] for long periods. Statistics indicates that idle devices account for 32% of total energy consumption. Leveraging these underutilized devices can help reduce resource waste and maximize efficiency.
- AI-Aware Edge Hardware Technology: The rapid advancement in edge hardware supports the deployment of edge AI inference. First, an increasing number of edge devices are now equipped with AI accelerators (e.g., NVIDIA NX Series [26]), enabling efficient inference. Second, the storage capacity of edge devices (e.g., home NAS [9] and Webcam [27]) has expanded significantly, with many now supporting TB-level storage. Third, the widespread adoption of 5G and Wi-Fi 6 [28] has significantly reduced the communication latency and increased the available bandwidth between edge devices and central servers, making edge inference more effective.
- Applications of Edge AI Inference: As deep learning models evolve, inference services have become critical for supporting many real-time applications (e.g., virtual reality [3], healthcare [4], and autonomous vehicles [5]). These applications require the timely processing of large volumes of data to deliver high-precision and low-latency intelligent services, where deploying AI models at edge has been widely believed to be a valid solution.

2.2 System Settings and Models

Our notations are summarized in Table 1.

Edge AI Inference System: As shown in Fig. 3, the edge AI inference system under our consideration is mainly composed of three entities. First, the inference service is operated and managed by a service provider, who wants to leverage the edge devices to deploy its AI models and run the inference service on such edge devices to serve the end users' inference queries. We use the set $\mathcal{M} = \{1, 2, ..., M\}$ to denote the AI models of the inference service. Second, the edge devices are owned by the corresponding device owners, respectively, and have available resources that can be used by the edge AI inference service. Such edge devices are often distributed at diverse locations in close proximity to the end users, and can include edge servers, desktops, and mobile devices. We use the set $\mathcal{N} = \{1, 2, ..., N\}$ to denote all the edge devices. Third, the end users submit inference queries to the edge AI inference service, which will be processed on the edge devices, and receive the inference results. We consider the entire system operating over consecutive time slots, denoted by the set $\mathcal{T} = \{1, 2, ..., T\}$.

Procurement Auctions: At each time slot, the edge AI inference service acts as the auctioneer and conducts an auction with the edge devices that act as the bidders. Without loss of generality, we consider each edge device submitting one bid in each auction. If an edge device does not submit a bid in an auction, we equivalently treat it as submitting a "virtual bid" with the positive infinity bidding price, and thus this virtual bid will never be chosen as a winning bid.

The auction at any time slot $t \in \mathcal{T}$ has multiple steps, as depicted in Fig. 3. First, the inference service solicits bids, and each edge device submits a bid in the format of $\mathcal{B}_{n,t} = \{b_{n,t}, c_n, \beta_n\}$, $n \in \mathcal{N}$, where $b_{n,t}$ represents the bidding price, i.e., the amount of money that the edge device n wants to charge for the inference service's usage of the resources on this edge device during the time slot t; c_n denotes the available resource capacity, e.g., total number of available processors (or processor cores) on the edge device n; and β_n is the processing speed per processor (or per processor core) on the edge device n. Here, processor types and speeds may vary across devices, but we only consider homogeneous processors (or cores) on a given device; our work in this paper can be easily extended to the heterogeneous case. Second, after receiving all the bids, the inference service determines the winning bids while also calculating the payment to each winning bid. Note that the payment to a winning bid may not be necessarily equal to the bidding price of the bid. Part of our goal in this paper is to design algorithms to decide the winning bids and the payments. Third, based on the auction outcome, the inference service sends selected AI models to the corresponding edge devices, dispatches the users' inference queries to those edge devices, uses the models to process the queries, and returns the inference results to the users. Fourth, the inference service makes payments to the edge devices. Depending on the reached agreement between the auctioneer and the bidders, the third step may occur before or after the fourth step.

Control Decisions: We consider multiple control decisions for the inference service at each time slot: $x_{n,t} \in \{1, 0\}$ indicates whether or not the inference service chooses the edge device (or the bid) n as a winner in the auction at the time slot t; $y_{n,m,t} \in \{1, 0\}$ indicates whether or not the inference service selects and deploys the model m on the edge device n at the time slot t; $z_{n,m,t} \in \mathbb{Z}_+$ indicates the number of inference queries that the inference service distributes to the model m on the edge device n for processing at the time slot t; and $w_{n,t} \geq 0$ indicates the payment that the inference service makes to the edge device n at the time slot t.

Cost of Inference Service: At each time slot t, the cost incurred at the inference service consists of multiple components. First, the service makes payments to the winning bids as $\sum_{n=1}^{N} x_{n,t} w_{n,t}$. Second, the service dispatches models to each edge device, which incurs the communication cost, denoted as $\sum_{n=1}^{N} \sum_{m=1}^{M} e_{n,m,t} y_{n,m,t} \delta_{n,m,t}$. Here, $e_{n,m,t}$ can represent the transmission time or traffic volume for sending the model m to the edge device n at the time slot t. $\delta_{n,m,t}$ is a binary indicator that captures whether or not the model m actually needs to be sent to the edge device n at the time slot t. If it is needed, then $\delta_{n,m,t} = 1$; if not, $\delta_{n,m,t} = 0$. The model may not be needed if this same model has already been on the edge device due to previous auctions; the model could be needed, because the model itself may be updated by the inference service, or the edge device may have deleted the model due to not winning a previous auction. Third, the service dispatches inference queries to each model on each edge device for processing, which also incurs the communication cost, denoted as $\sum_{n=1}^{N} \sum_{m=1}^{M} d_{n,t} z_{n,m,t}$. Here, $d_{n,t}$ can represent the transmission time or traffic volume for sending a single inference query to the edge device n at the time slot t. Consequently, the total cost of the inference service over time is $\sum_{t=1}^{T} (\sum_{n=1}^{N} x_{n,t} w_{n,t} + \sum_{n=1}^{N} \sum_{m=1}^{M} (e_{n,m,t} y_{n,m,t} \delta_{n,m,t} + d_{n,t} z_{n,m,t}))$.

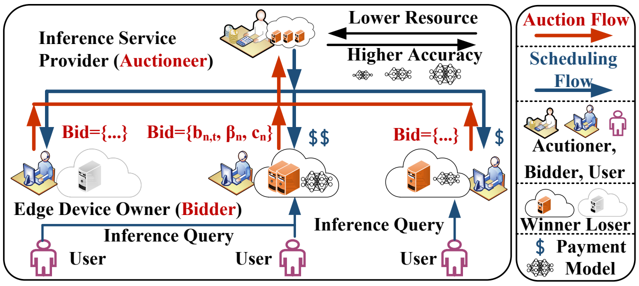


Fig. 3: Auction in the Edge AI Inference System

Cost of Edge Devices: At each time slot t, the cost incurred on each edge device n also consists of multiple components. First, the bidding cost $x_{n,t} b_{n,t}$, which can be determined by the edge device's operational cost such as electricity consumption at the time slot t, amortized hardware expense per time slot, etc. Second, the payment received from the inference service, which is treated as negative cost, i.e., $-x_{n,t} w_{n,t}$. The third is the "switching cost". When the inference service did not choose an edge device as a winner in the previous auction at $t-1$ but chooses this edge device as a winner in the current auction at t, then the inference service may need to re-connect to and re-authorize the edge device; and the edge device may also need to re-initialize the execution environment, re-prepare the resources, or even re-start the device. Such operations can take the leading time, considered as a type of cost that we call the switching cost. We use $\sum_{n=1}^{N} l_n [x_{n,t} - x_{n,t-1}]^+$ to denote the switching cost, where $[\cdot]^+ \triangleq \max\{\cdot, 0\}$, and l_n is such leading or startup time of the edge device n. Consequently, the total cost of the edge devices over time is $\sum_{t=1}^{T} \sum_{n=1}^{N} (x_{n,t}(b_{n,t} - w_{n,t}) + l_n [x_{n,t} - x_{n,t-1}]^+)$.

Error Rate of Inference: If we use $\Upsilon_{n,m,t}$ to represent the accuracy of the model m on the edge n at the time slot t, then we can define the error rate correspondingly as $a_{n,m,t} = 1 - \Upsilon_{n,m,t}$. That is, the lower the error rate is, the better the service quality the inference service provides. The error rate depends on the specific model chosen and the specific inference queries that are resolved by this model [29]. A model may also have different versions with different error rates, which can be essentially treated as different models. The total error rate of the inference of all models on all edge devices over time is $\sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} a_{n,m,t} y_{n,m,t}$. Note that $a_{n,m,t}$ is posterior, i.e., it is only observed by the inference service after deploying the model m on the edge n and using this model to resolve inference queries there at the time slot t, no matter what value it is. As the inference queries from the end users could vary as time goes, the error rate of a given model could thus also vary.

2.3 Problem Formulation

Optimization Constraints: We consider the following constraints for our optimization problem.

First, on each edge n, the AI models deployed there must respect the resource capacity of the edge:

$$\forall t, n : \textstyle\sum_{m=1}^{M} r_m y_{n,m,t} - c_n x_{n,t} \le 0, \tag{a}$$

where r_m is the amount of resource, e.g., the number of processors (or processor cores) requested by the model m for conducting inference; and c_n is as explained before.

Second, each edge device n maintains a queue formed by the inference queries and the queue dynamics needs to be correctly reflected:

$$\begin{aligned}\forall t, n : Q_{t+1,n} = [Q_{t,n} + \textstyle\sum_m z_{n,m,t} - \sum_m y_{n,m,t} p_{n,m}]^+,\\ \forall t, n : Q_{t,n} \le q_n x_{n,t}.\end{aligned} \tag{b}$$

Here, $Q_{t,n}$, $\forall t$ is the current length of the queue, i.e., the number of inference queries that are currently in the queue and waiting to be processed, on the edge device n at the time slot t; $z_{n,m,t}$ refers to the inference queries that arrive at the edge device n to be processed by the AI model m at the time slot t; $p_{n,m}$ refers to the number of inference queries processed by the model m on the edge device n per time slot, which could depend on β_n, r_m, and the model m itself; and q_n refers to the query queue capacity on the edge device n. We enforce $Q_{T+1,n} = 0$, $\forall n$ i.e., the queue on every device is eventually cleared. We use the function $[\cdot]^+ \triangleq \max\{\cdot, 0\}$ to capture the transition from $Q_{t,n}$ to $Q_{t+1,n}$, $\forall t, n$. Note that we target edge AI services that use their models to process "small" and lightweight inference queries quickly and efficiently. We enforce the queue capacity for each time slot, by which we implicitly push the queued inference queries to get processed. We do not consider lengthy inference tasks where a task may need to run for multiple time slots, and also do not consider explicit deadlines in this work.

Third, all the inference queries submitted to the inference service at the time slot t, i.e., τ_t, need to be fully dispatched to all the AI models on all the edge devices:

$$\forall t : \textstyle\sum_{m=1}^{M} \sum_{n=1}^{N} z_{n,m,t} = \tau_t. \tag{c}$$

Optimization Problem: We minimize the *social cost* over time, i.e., the sum of the total cost of the inference service, the total cost of the edge devices, and the total error rate of the inference over time, subject to Constraints (a)∼(c). Yet, solving this optimization problem as is in an online manner is particularly challenging, if ever possible. First, some inputs cannot be observed in time and can only be revealed after making the decisions, e.g., the error rate $a_{m,t}$, and/or the transmission time $e_{n,m,t}$ and $d_{n,t}$. Making control decisions obliviously without observing the inputs is intrinsically difficult. Second, the state-transition constraint based on the non-linear operator $[\cdot]^+$ couples every pair of adjacent time slots. It is non-trivial to enforce such queue dynamics on the fly while eventually clearing the queue.

We thus reformulate our problem as follows, and treat this reformulated problem as our original problem $\mathscr{P}$. In this reformulation, we selectively relax the "instantaneous" Constraints (b)∼ (c) to the "long-term" Constraints $C_2 \sim C_4$. That is, rather than enforcing the instantaneous constraint at every time slot, now we only enforce such constraints cumulatively in the long run.

$$\begin{aligned}
&\min \textstyle\sum_{t=1}^{T}\{\sum_{n=1}^{N}\sum_{m=1}^{M}\{e_{n,m,t}y_{n,m,t}\delta_{n,m,t} + a_{n,m,t}y_{n,m,t} + \\
&\qquad d_{n,t}z_{n,m,t}\} + \textstyle\sum_{n=1}^{N}\{x_{n,t}b_{n,t} + l_n[x_{n,t} - x_{n,t-1}]^+\}\}\\
&\text{s.t. } C_1 : \forall t, n : h_t^n = \sum_{m=1}^{M} r_m y_{n,m,t} - c_n x_{n,t} \le 0,\\
&C_2 : \forall n : \sum_{t=1}^{T} g_{n,t}^1 = \sum_{t=1}^{T}((T-t)(\sum_{m=1}^{M}(z_{n,m,t} - y_{n,m,t}p_{n,m}))\\
&\qquad - q_n x_{n,t}) \le 0,\\
&C_3 : \sum_{t=1}^{T} g_t^2 = \sum_{t=1}^{T}(\sum_{n=1}^{N}\sum_{m=1}^{M} z_{n,m,t} - \tau_t) \le 0,\\
&C_4 : \sum_{t=1}^{T} g_t^3 = \sum_{t=1}^{T}(\tau_t - \sum_{n=1}^{N}\sum_{m=1}^{M} z_{n,m,t}) \le 0,\\
&\text{var. } \forall t, n, m : x_{n,t} \in \{0, 1\}, y_{n,m,t} \in \{0, 1\}, z_{n,m,t} \in \mathbb{Z}_+.
\end{aligned} \tag{1}$$

Constraint C_1 is Constraint (a), ensuring the resource capacity on each edge at every time slot. Constraint C_2 is from Constraint (b). From Constraint (b), for each n, we have $Q_{T+1,n} \ge Q_{T,n} + \sum_m z_{n,m,T} - \sum_m y_{n,m,T}p_{n,m} \ge Q_{T-1,n} + \sum_{t=T-1}^{T}(\sum_m z_{n,m,t} - \sum_m y_{n,m,t}p_{n,m}) \ge \ldots \ge Q_{1,n} + \sum_{t=1}^{T}(\sum_m z_{n,m,t} - \sum_m y_{n,m,t}p_{n,m}) \ge \sum_{t=1}^{T}(\sum_m z_{n,m,t} - \sum_m y_{n,m,t}p_{n,m})$, where all "$\ge$" hold since $[a]^+ \ge a$, for any a, and the last "$\ge$" holds due to $Q_{0,n} = 0$, $z_{n,m,0} = 0$ and $y_{n,m,0} = 0$, $\forall n, m$. Also, due to $Q_{T+1,n} = 0$, we then have $\forall t, q_n x_{n,t+1} \ge Q_{t+1,n} \ge \sum_{t'=1}^{t}(\sum_m z_{n,m,t'} - \sum_m y_{n,m,t'}p_{n,m})$. Then we sum the inequalities above from $t = 1$ to $T - 1$ and obtain Constraint C_2. Constraints C_3 and C_4 are from Constraint (c). In the above, the optimization objective is to minimize the social cost. Aligned with existing typical social cost optimization, the payments are cancelled in the social cost; yet, we still need to calculate the payments as part of our auction outcome in each time slot. Such cancellation is a common practice in lots of existing auction-related research. Yet, this does not imply that $w_{n,t}$ is independent of other decision variables.

Algorithmic Challenges: Solving the relaxed problem $\mathscr{P}$ as above in an online manner is still challenging. First, the switching cost that couples adjacent time slots still exists in the objective, so no matter what the bid-selection decision is made currently, it will impact the switching cost between the current time slot and the next time slot where the bid-selection decision for the next time slot is yet unknown currently. Second, the long-term constraints, supposed to be easier than instantaneous constraints though, also need to be carefully handled and enforced on the fly and due to uncertain future conditions and posterior observed inputs. Third, the integer decision variables make our problem an integer program, which is typically NP-hard [30], even in the offline setting. Fourth, we need to determine the payment for each winning bid in each auction, which often depends on the bidding prices and the auction outcome,

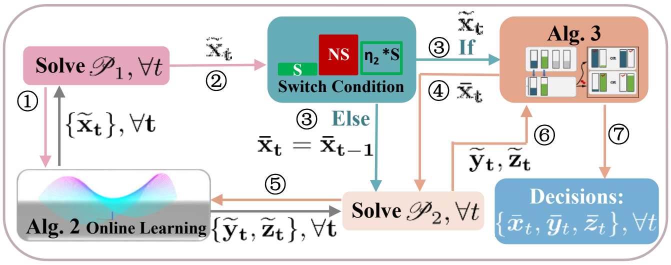

Solve $\mathscr{P}_1, \forall t$
$\widetilde{\mathbf{x}}_\mathbf{t}$
②
①
$\{\widetilde{\mathbf{x}}_\mathbf{t}\}, \forall \mathbf{t}$
S
NS
η₂ *S
Switch Condition
$\widetilde{\mathbf{x}}_\mathbf{t}$
③ If
Alg. 3
④ $\bar{\mathbf{x}}_\mathbf{t}$
③ Else
$\bar{\mathbf{x}}_\mathbf{t} = \bar{\mathbf{x}}_{\mathbf{t-1}}$
⑥
⑦
$\widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}$
⑤
Alg. 2 Online Learning
$\{\widetilde{\mathbf{y}}_\mathbf{t}, \widetilde{\mathbf{z}}_\mathbf{t}\}, \forall \mathbf{t}$
Solve $\mathscr{P}_2, \forall t$
Decisions:
$\{\bar{x}_t, \bar{y}_t, \bar{z}_t\}, \forall t$

Fig. 4: Workflow of the Proposed Approach

in order to satisfy the desired economic properties such as truthfulness and individual rationality as elaborated later.

3 Online Mechanism Design

Algorithm 1 is our main online control algorithm, which invokes Algorithm 2 and Algorithm 3 for bid selection, model deployment, and query dispatch and invokes Algorithm 4 for payment allocation. This is shown in Fig. 4. These four algorithms are elaborated in Sections 3.1∼3.4, respectively, where we also highlight our insights on how each of our algorithms has overcome the various aforementioned algorithmic challenges. Our proposed algorithms compose a creative and innovative combination that utilizes and unifies different ideas in a common framework of handling NP-hardness in polynomial time in repetitive online auctions with switching cost and long-term constraints, while being able to provide provable overall performance guarantees.

Notations: We use $\{\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}}\}$ to denote the integral solution returned by Algorithm 1 as it invokes Algorithm 3, where $\{\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t\}$ is for each time slot t. We also use $\{\widetilde{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t\}$, $\forall t$ to denote the fractional solution produced within Algorithm 1 as it invokes Algorithm 2.

3.1 Controlled Switch

We decouple Problem $\mathscr{P}$ into two sub-problems, i.e., Problem $\mathscr{P}_1$ and Problem $\mathscr{P}_2$, based on the switching cost. To that end, we need to introduce some auxiliary notations:

$$\Theta_S^t(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \textstyle\sum_{n=1}^N l_n[x_{n,t} - x_{n,t-1}]^+,$$
$$\Theta_{-S}^t(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t) = \textstyle\sum_{n=1}^N \sum_{m=1}^M (e_{n,m,t} y_{n,m,t} \delta_{n,m,t} +$$
$$a_{n,m,t} y_{n,m,t} + d_{n,t} z_{n,m,t}) + \textstyle\sum_{n=1}^N x_{n,t} b_{n,t}.$$
$$\Theta^t = \Theta_S^t(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) + \Theta_{-S}^t(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t).$$

Having these, Problem $\mathscr{P}_1$ and Problem $\mathscr{P}_2$ are as follows.

$$\begin{aligned}
&\min \textstyle\sum_{t=1}^T \mathcal{P}_{t,1} \triangleq \sum_{t=1}^T \Theta_{-S}^t(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)\\
&s.t. \quad \forall t : \boldsymbol{h}_t^1 \triangleq \boldsymbol{h}_t(\boldsymbol{x}_t, \boldsymbol{y}_t) \preceq \boldsymbol{0},\\
&\qquad \textstyle\sum_{t=1}^T \boldsymbol{g}_t^1(\boldsymbol{y}_t, \boldsymbol{z}_t) \preceq \boldsymbol{0}, \sum_{t=1}^T g_t^2(\boldsymbol{z}_t) \preceq \boldsymbol{0}, \sum_{t=1}^T g_t^3(\boldsymbol{z}_t) \preceq \boldsymbol{0},\\
&\text{var.} \quad \forall t, n, m : x_{n,t} \in [0, 1], y_{n,m,t} \in [0, 1], z_{n,m,t} \in \mathbb{R}_+.
\end{aligned}$$

$$\begin{aligned}
&\min \textstyle\sum_{t=1}^T \mathcal{P}_{t,2} \triangleq \sum_{t=1}^T \Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)\\
&s.t. \quad \forall t : \boldsymbol{h}_t^2 \triangleq \boldsymbol{h}_t(\bar{\boldsymbol{x}}_t, \boldsymbol{y}_t) \preceq \boldsymbol{0},\\
&\qquad \textstyle\sum_{t=1}^T \boldsymbol{g}_t^1(\boldsymbol{y}_t, \boldsymbol{z}_t) \preceq \boldsymbol{0}, \sum_{t=1}^T g_t^2(\boldsymbol{z}_t) \preceq \boldsymbol{0}, \sum_{t=1}^T g_t^3(\boldsymbol{z}_t) \preceq \boldsymbol{0},\\
&\text{var.} \quad \forall t, n, m : y_{n,m,t} \in [0, 1], z_{n,m,t} \in \mathbb{R}_+.
\end{aligned}$$

Note that, compared to $\mathscr{P}_1$, the decision variables are only $\boldsymbol{y}$ and $\boldsymbol{z}$ in $\mathscr{P}_2$; and $\bar{\boldsymbol{x}}$, produced by our algorithms, is given as input in $\mathscr{P}_2$. Here, $\boldsymbol{h}_t$ is an aggregation of $h_t^1, h_t^2, \ldots$, and h_t^N; and $\boldsymbol{g}_t^1$ is an aggregation of $g_{1,t}^1, g_{2,t}^1, \ldots$, and $g_{N,t}^1$.

Algorithm 1 Online Auction Algorithm (OAA)

```
1:  Initialization: $t' = 0, \bar{\boldsymbol{x}}_{-1} = 0, \bar{\boldsymbol{x}}_0 = \bar{\boldsymbol{y}}_0 = \bar{\boldsymbol{z}}_0 = 0$ ;
2:  for $t = 1, 2, \ldots, T$ do
3:      Obtain $\widetilde{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t$ from $\mathscr{P}_1$ via Algorithm 2;
4:      if $\Theta_S^{t'}(\bar{\boldsymbol{x}}_{t'}, \bar{\boldsymbol{x}}_{t'-1}) \leq \eta \sum_{u=t'}^{t-1} \Theta_{-S}^u(\bar{\boldsymbol{x}}_u, \bar{\boldsymbol{y}}_u, \bar{\boldsymbol{z}}_u)$ then
5:          Round $\widetilde{\boldsymbol{x}}_t$ to $\bar{\boldsymbol{x}}_t$ via Algorithm 3;
6:          if $\bar{\boldsymbol{x}}_t \neq \bar{\boldsymbol{x}}_{t-1}$ then
7:              Set $t' = t$;
8:          end if
9:      end if
10: Set $\bar{\boldsymbol{x}}_t = \bar{\boldsymbol{x}}_{t'}$, and get $\widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t$ from $\mathscr{P}_2$ via Algorithm 2;
11: Round $\widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t$ to $\bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t$ via Algorithm 3;
12: Apply decisions $\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t$ to $t$;
13: Invoke Algorithm 4 to allocate payment for $t$;
14: end for
```

Algorithm 1 uses the pre-specified parameter $\eta > 0$ to balance the switching cost and the non-switching cost. In Line 3, the fractional decisions of winning-bid selection, model placement, and query dispatch are computed, via Algorithm 2 which will be elaborated in Section 3.2. In Line 4, if the cumulative non-switching cost since the time slot t' exceeds $\frac{1}{\eta}$ times the switching cost of the last switch operation that occurred from $t' - 1$ to t', then it seeks to use the new decisions which have been just computed, as in Lines 5∼8. In Line 5, the fractional decisions are rounded into integers using Algorithm 3, which will also be elaborated later. Lines 6∼8 record the time slot as a new switch operation has to occur. In Line 10, it uses $\bar{\boldsymbol{x}}_t$ as the given input to solve $\mathscr{P}_2$ via Algorithm 2, where $\bar{\boldsymbol{x}}_t$ can be either just generated at the current time slot or carried over from the previous time slot, depending on the value of t' at this point. In Line 11, all other corresponding decisions are rounded. Line 13 calculates the payment via Algorithm 4, which will be described later.

Insights: The way the parameter η is used in Algorithm 1 forces that between adjacent switch operations the previous switching cost is no greater than η times the non-switching cost accumulated; intuitively, this guarantees the total cost, including both switching and non-switching, is no greater than $1 + \eta$ times the non-switching cost. To further make such total cost incurred by our online decisions "competitive" against the offline optimum, we only need to focus on the offline optimum of the non-switching cost, eventually removing our concern on the switching cost. This "wait and delay" idea on handling switch operations is passive or reactive, rather than proactive; yet, it is well under pre-specified control via η. This enables us to overcome the dependency between bid selections in adjacent auctions.

3.2 Online Learning

We note that both $\mathscr{P}_1$ and $\mathscr{P}_2$ are in the form as follows:

$$\begin{aligned}
&\min_{\mathcal{X}_t \in \mathbb{X}} \quad \textstyle\sum_{t=1}^T f_t(\mathcal{X}_t)\\
&\text{s.t.} \qquad h_t(\mathcal{X}_t) \preceq \boldsymbol{0}, \forall t; \textstyle\sum_{t=1}^T g_t(\mathcal{X}_t) \preceq \boldsymbol{0},
\end{aligned}$$

where $\mathcal{X}_t$ is the decision variable for each t and $\mathbb{X}$ is the domain; $\sum_{t=1}^T f_t(\cdot)$ is the objective function, subject to the long-term constraint $\sum_{t=1}^T g_t(\cdot) \preceq \boldsymbol{0}$ over the time horizon,

Algorithm 2 Online Learning Algorithm (OLA), $\forall t$

```
// For $\mathscr{P}_1$,$\mathcal{X}_t$=$\{\widetilde{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t\}$, $f_t = \mathcal{P}_{t,1}$, $h_t = \boldsymbol{h}_t^1, \forall t$;
// For $\mathscr{P}_2$,$\mathcal{X}_t$=$\{\widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t\}$, $f_t = \mathcal{P}_{t,2}$, $h_t = \boldsymbol{h}_t^2, \forall t$;
// $g_t = \{\boldsymbol{g}_t^1, g_t^2, g_t^3\}, \forall t$;
// Initialize step sizes $\mu$ and $\alpha$;
1: Set $\boldsymbol{\omega}_t = [\boldsymbol{\omega}_{t-1} + \mu g_{t-1}(\mathcal{X}_{t-1})]^+$;
2: Given $\boldsymbol{\omega}_t$, obtain $\mathcal{X}_t$ by solving the problem below:
       $\min_{\mathcal{X} \in \mathbb{X}} \nabla f_{t-1}(\mathcal{X}_{t-1})(\mathcal{X} - \mathcal{X}_{t-1}) + \boldsymbol{\omega}_t g_{t-1}(\mathcal{X}) + \frac{||\mathcal{X} - \mathcal{X}_{t-1}||^2}{2\alpha}$
       s.t. $h_{t-1}(\mathcal{X}) \preceq \mathbf{0}$.
3: Observe $f_t$, $g_t$, and $h_t$ to be used at $t+1$;
```

and the instantaneous constraint $h(\cdot) \preceq \mathbf{0}$ for each t. This problem can be reformulated equivalently using the Lagrangian method [31], [32] into the following form:

$$
\begin{aligned}
&\min\nolimits_{\mathcal{X}_t \in \mathbb{X}} \max\nolimits_{\boldsymbol{\omega}_t \in \mathbb{R}_{\geq 0}^{dim(\mathcal{X}_t)}} && \textstyle\sum_{t=1}^{T} \Big(f_t(\mathcal{X}_t) + \boldsymbol{\omega}_t g_t(\mathcal{X}_t) \Big), \\
&\text{s.t.} && h_t(\mathcal{X}_t) \preceq \mathbf{0}, \forall t,
\end{aligned}
$$

where $\boldsymbol{\omega}_t$ represents the Lagrange multiplier. Based on this, we can then design a primal-dual method to update the primal variable $\mathcal{X}_t$ and the dual variable $\boldsymbol{\omega}_t$ alternately in an online manner as time goes.

Algorithm 2 does exactly this. That is, at each time slot t, it firstly computes $\boldsymbol{\omega}_t$ in Line 1, where $\mu \geq 0$ is the step size. Then, in Line 2, given $\boldsymbol{\omega}_t$, it solves that problem and uses its solution as $\mathcal{X}_t$, where $\frac{||\mathcal{X} - \mathcal{X}_{t-1}||^2}{2\alpha}$ is a regularization term. Thus, note that these are not standard primal-dual steps, but have a "modified" gradient-descent step. We will show later that we can obtain provable performance this way. Also, in Line 2, this per-time-slot problem is a standard convex optimization problem [33], and thus can be solved by directly invoking any standard convex optimization solver (e.g., CVXPY [34] or Gurobi [35]).

Insights: We highlight the following two insights with Algorithm 2. First, the long-term constraint is no longer a concern, because it is absorbed into the objective and then only the instantaneous constraint still exists, naturally making the problem splittable and solvable at each individual time slot. Second, the posterior input is also no longer a challenge, because, for example, when computing $\mathcal{X}_t$, no input about or beyond the time slot t is needed or used, not to mention anything posterior. In fact, this is why this process can be called "online learning". These two insights enable us to overcome the challenges of the query queue dynamics and the posterior inference error rate.

3.3 Randomized Rounding

Algorithm 3 rounds the fractional decisions into integers, as all control decisions in the original problem $\mathscr{P}$ are integers.

To round and $\widetilde{\boldsymbol{z}}_t$ and $\widetilde{\boldsymbol{x}}_t$, as in Lines 19∼22 and Lines 23∼26, respectively, we use a simple randomized approach which utilizes their fractional parts as the probabilities to round them into integers. Our goal is to ensure $E(\bar{\boldsymbol{z}}_t) = \widetilde{\boldsymbol{z}}_t$ and $E(\bar{\boldsymbol{x}}_t) = \widetilde{\boldsymbol{x}}_t$. $\widetilde{\boldsymbol{z}}$ and $\widetilde{\boldsymbol{x}}$ appear in the long-term constraints of the problem $\mathscr{P}$. We allow but will bound the violation of such long-term constraints. Both $E(\bar{\boldsymbol{z}}_t) = \widetilde{\boldsymbol{z}}_t$ and $E(\bar{\boldsymbol{x}}_t) = \widetilde{\boldsymbol{x}}_t$ are important for the bound analysis; the latter is also crucial for the economic properties analysis.

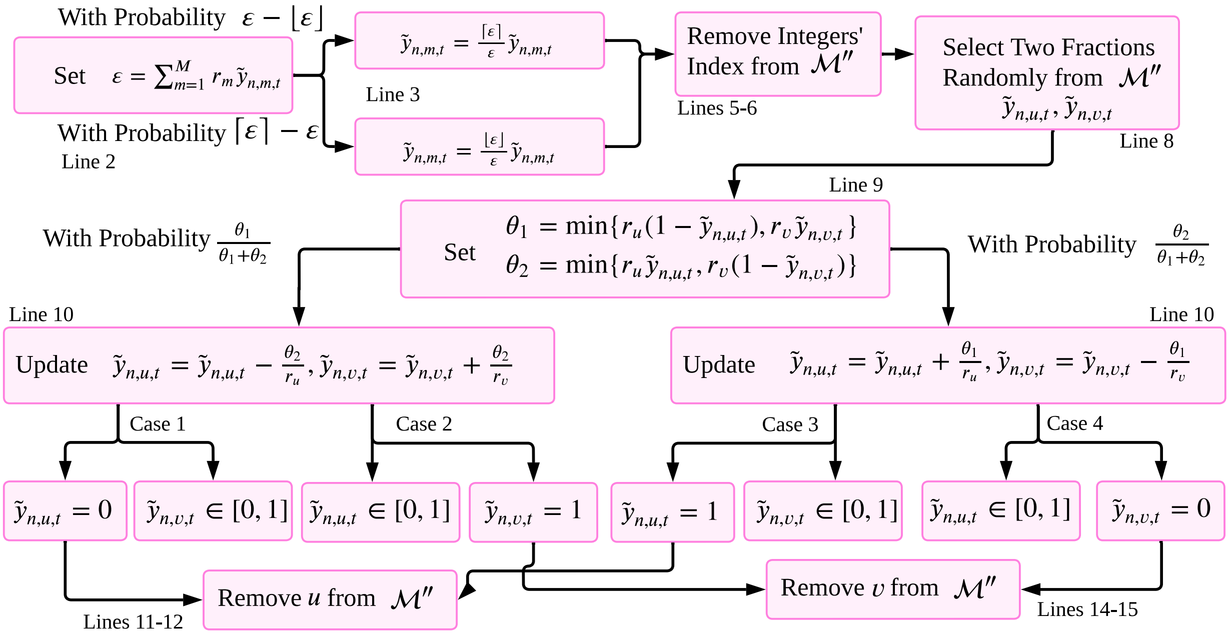
Algorithm 3 Randomized Rounding Algorithm (RRA), $\forall t$

```
Input: The fractional solutions $\tilde{\boldsymbol{y}}_t$ or $\tilde{\boldsymbol{z}}_t$ or $\tilde{\boldsymbol{x}}_t$;
Output: The integer solutions $\bar{\boldsymbol{y}}_t$ or $\bar{\boldsymbol{z}}_t$ or $\bar{\boldsymbol{x}}_t$;
 // Round $\tilde{\boldsymbol{y}}_t$
1: for $n \in \mathcal{N}$ do
2:     if $\varepsilon = \sum_{m=1}^{M} r_m \tilde{y}_{n,m,t}$ is not an integer then
3:         With probability $\varepsilon - \lfloor \varepsilon \rfloor$, set $\tilde{y}_{n,m,t} = \frac{\lceil \varepsilon \rceil}{\varepsilon} \tilde{y}_{n,m,t}, \forall m$;
           With probability $\lceil \varepsilon \rceil - \varepsilon$, set $\tilde{y}_{n,m,t} = \frac{\lfloor \varepsilon \rfloor}{\varepsilon} \tilde{y}_{n,m,t}, \forall m$;
4:     end if
5:     Set $\mathcal{M}' = \{m | \tilde{y}_{n,m,t} \in \{0,1\}\}$, $\mathcal{M}'' = \mathcal{M} \backslash \mathcal{M}'$;
6:     Set $\bar{y}_{n,m,t} = \tilde{y}_{n,m,t}, \forall m \in \mathcal{M}'$;
7:     while $|\mathcal{M}''| \geq 0$ do
8:         Choose $u, v \in \mathcal{M}''$, where $u \neq v$;
9:         Set $\theta_1 = \min\{r_u(1 - \tilde{y}_{n,u,t}), r_v \tilde{y}_{n,v,t}\}$,
           Set $\theta_2 = \min\{r_u \tilde{y}_{n,u,t}, r_v(1 - \tilde{y}_{n,v,t})\}$;
10:        With probability $\frac{\theta_2}{\theta_1 + \theta_2}$,
           Set $\tilde{y}_{n,u,t} = \tilde{y}_{n,u,t} + \frac{\theta_1}{r_u}$, $\tilde{y}_{n,v,t} = \tilde{y}_{n,v,t} - \frac{\theta_1}{r_v}$;
           With probability $\frac{\theta_1}{\theta_1 + \theta_2}$,
           Set $\tilde{y}_{n,u,t} = \tilde{y}_{n,u,t} - \frac{\theta_2}{r_u}$, $\tilde{y}_{n,v,t} = \tilde{y}_{n,v,t} + \frac{\theta_2}{r_v}$;
11:        if $\tilde{y}_{n,u,t} \in \{0,1\}$ then
12:            Set $\bar{y}_{n,u,t} = \tilde{y}_{n,u,t}$, $\mathcal{M}'' = \mathcal{M}'' \backslash \{u\}$;
13:        end if
14:        if $\tilde{y}_{n,v,t} \in \{0,1\}$ then
15:            Set $\bar{y}_{n,v,t} = \tilde{y}_{n,v,t}$, $\mathcal{M}'' = \mathcal{M}'' \backslash \{v\}$;
16:        end if
17:    end while
18: end for
 // Round $\tilde{\boldsymbol{z}}_t$
19: Set $\bar{z}_{n,m,t} = \tilde{z}_{n,m,t}, \forall (n,m) \in \{(n,m) | \tilde{z}_{n,m,t} \in \mathbb{Z}_+\}$;
20: for $(n,m) \in \mathcal{N} \times \mathcal{M} \backslash \{(n,m) | \tilde{z}_{n,m,t} \in \mathbb{Z}_+\}$ do
21:     With probability $\tilde{z}_{n,m,t} - \lfloor \tilde{z}_{n,m,t} \rfloor$,set $\bar{z}_{n,m,t} = \lceil \tilde{z}_{n,m,t} \rceil$;
        With probability $\lceil \tilde{z}_{n,m,t} \rceil - \tilde{z}_{n,m,t}$,set $\bar{z}_{n,m,t} = \lfloor \tilde{z}_{n,m,t} \rfloor$;
22: end for
 // Round $\tilde{\boldsymbol{x}}_t$
23: Set $\bar{x}_{n,t} = \tilde{x}_{n,t}, \forall n \in \{n | \tilde{x}_{n,t} \in \{0,1\}\}$;
24: for $n \in \mathcal{N} \backslash \{n | \tilde{x}_{n,t} \in \{0,1\}\}$ do
25:     With probability $\tilde{x}_{n,t} - \lfloor \tilde{x}_{n,t} \rfloor$, set $\bar{x}_{n,t} = \lceil \tilde{x}_{n,t} \rceil$;
        With probability $\lceil \tilde{x}_{n,t} \rceil - \tilde{x}_{n,t}$, set $\bar{x}_{n,t} = \lfloor \tilde{x}_{n,t} \rfloor$;
26: end for
```

To see $E(\bar{\boldsymbol{x}}_t) = \widetilde{\boldsymbol{x}}_t$, for example, we have $E(\bar{x}_{n,t}) = (\tilde{x}_{n,t} - \lfloor \tilde{x}_{n,t} \rfloor) \lceil \tilde{x}_{n,t} \rceil + (\lceil \tilde{x}_{n,t} \rceil - \tilde{x}_{n,t}) \lfloor \tilde{x}_{n,t} \rfloor = (\lceil \tilde{x}_{n,t} \rceil - \lfloor \tilde{x}_{n,t} \rfloor) \tilde{x}_{n,t} = \tilde{x}_{n,t}$.

To round $\widetilde{\boldsymbol{y}}_t$, as in Lines 1∼18, we design a more sophisticated double randomization approach. The goal here is multi-fold. We elaborate on them below, respectively. We also provide a flowchart here, as shown in Fig. 5.

First, we ensure the instantaneous constraint of the problem $\mathscr{P}$, where $\widetilde{\boldsymbol{y}}$ appears, has no violation after rounding $\widetilde{\boldsymbol{y}}$. Note that, for each n at each t, before rounding $\widetilde{\boldsymbol{y}}$, we have $\varepsilon = \sum_m r_m \tilde{y}_{n,m,t} \leq c_n \bar{x}_{n,t}$. If $\bar{x}_{n,t} = 0$, then we directly have $\bar{y}_{n,m,t} = \tilde{y}_{n,m,t} = 0, \forall m$. If $\bar{x}_{n,t} = 1$, then after Line 3, we either have $\frac{\lceil \varepsilon \rceil}{\varepsilon} \sum_m r_m \tilde{y}_{n,m,t} = \frac{\lceil \varepsilon \rceil}{\varepsilon} \varepsilon = \lceil \varepsilon \rceil \leq c_n$ as c_n is an integer, or have $\frac{\lfloor \varepsilon \rfloor}{\varepsilon} \sum_m r_m \tilde{y}_{n,m,t} = \frac{\lfloor \varepsilon \rfloor}{\varepsilon} \varepsilon = \lfloor \varepsilon \rfloor \leq c_n$. That is, the constraint still holds at this point. As we continue to Line 10, for the arbitrary u and v, we either have $r_u(\tilde{y}_{n,u,t} + \frac{\theta_1}{r_u}) + r_v(\tilde{y}_{n,v,t} - \frac{\theta_1}{r_v}) = r_u \tilde{y}_{n,u,t} + r_v \tilde{y}_{n,v,t}$, or have $r_u(\tilde{y}_{n,u,t} - \frac{\theta_2}{r_u}) + r_v(\tilde{y}_{n,v,t} + \frac{\theta_2}{r_v}) = r_u \tilde{y}_{n,u,t} + r_v \tilde{y}_{n,v,t}$. That is, after this step, the sum between any $r_u \tilde{y}_{n,u,t}$ and



With Probability $\varepsilon - \lfloor\varepsilon\rfloor$
Set $\varepsilon = \sum_{m=1}^{M} r_m \tilde{y}_{n,m,t}$
With Probability $\lceil\varepsilon\rceil - \varepsilon$
Line 2
$\tilde{y}_{n,m,t} = \frac{\lceil\varepsilon\rceil}{\varepsilon}\tilde{y}_{n,m,t}$
Line 3
$\tilde{y}_{n,m,t} = \frac{\lfloor\varepsilon\rfloor}{\varepsilon}\tilde{y}_{n,m,t}$
Remove Integers' Index from $\mathcal{M}''$
Lines 5-6
Select Two Fractions Randomly from $\mathcal{M}''$ $\tilde{y}_{n,u,t}, \tilde{y}_{n,v,t}$
Line 8
Line 9
With Probability $\frac{\theta_1}{\theta_1+\theta_2}$
Set $\theta_1 = \min\{r_u(1-\tilde{y}_{n,u,t}), r_v\tilde{y}_{n,v,t}\}$ $\theta_2 = \min\{r_u\tilde{y}_{n,u,t}, r_v(1-\tilde{y}_{n,v,t})\}$
With Probability $\frac{\theta_2}{\theta_1+\theta_2}$
Line 10
Update $\tilde{y}_{n,u,t} = \tilde{y}_{n,u,t} - \frac{\theta_2}{r_u}, \tilde{y}_{n,v,t} = \tilde{y}_{n,v,t} + \frac{\theta_2}{r_v}$
Line 10
Update $\tilde{y}_{n,u,t} = \tilde{y}_{n,u,t} + \frac{\theta_1}{r_u}, \tilde{y}_{n,v,t} = \tilde{y}_{n,v,t} - \frac{\theta_1}{r_v}$
Case 1
Case 2
Case 3
Case 4
$\tilde{y}_{n,u,t} = 0$
$\tilde{y}_{n,v,t} \in [0,1]$
$\tilde{y}_{n,u,t} \in [0,1]$
$\tilde{y}_{n,v,t} = 1$
$\tilde{y}_{n,u,t} = 1$
$\tilde{y}_{n,v,t} \in [0,1]$
$\tilde{y}_{n,u,t} \in [0,1]$
$\tilde{y}_{n,v,t} = 0$
Remove u from $\mathcal{M}''$
Lines 11-12
Remove v from $\mathcal{M}''$
Lines 14-15

Fig. 5: Flowchart for Randomized Rounding

$r_v\tilde{y}_{n,v,t}$ does not change. So, the instantaneous constraint still holds.

Second, we ensure $E(\bar{\boldsymbol{y}}_t) = \tilde{\boldsymbol{y}}_t$. In Line 3, we have $(\varepsilon - \lfloor\varepsilon\rfloor)\frac{\lceil\varepsilon\rceil}{\varepsilon}\tilde{y}_{n,m,t}+(\lceil\varepsilon\rceil-\varepsilon)\frac{\lfloor\varepsilon\rfloor}{\varepsilon}\tilde{y}_{n,m,mt} = \tilde{y}_{n,m,t}$; further, in Line 10, for u, we have $\frac{\theta_2}{\theta_1+\theta_2}(\tilde{y}_{n,u,t}+\frac{\theta_1}{r_u}) + \frac{\theta_1}{\theta_1+\theta_2}(\tilde{y}_{n,u,t}-\frac{\theta_2}{r_u}) = \tilde{y}_{n,u,t}$, and for v, have the analogous result. That is, after taking the expectation twice, we have $E(\bar{\boldsymbol{y}}_t) = \tilde{\boldsymbol{y}}_t$, which also contributes to the violation bound analysis of the long-term constraints.

Third, we ensure each iteration of Lines 7∼17 produces at least one integer. In Line 9, no matter what values θ_1 and θ_2 take, Line 10 always produces at least one integer. For example, suppose $\theta_1 = r_u(1-\tilde{y}_{n,u,t})$ and $\theta_2 = r_u\tilde{y}_{n,u,t}$. Then, in the first case of Line 10, we have $\tilde{y}_{n,u,t} = 1$ and in the second case of Line 10, we have $\tilde{y}_{n,u,t} = 0$. One can easily verify Line 10 for all the other cases of θ_1 and θ_2. This shows the conversion from fractions to integers succeeds.

Insights: Algorithm 3, with Algorithm 2 in the framework of Algorithm 1, overcomes the NP-hardness of our problem, because the relaxed problem without integer decision variables is not NP-hard any more but polynomial-time solvable; and rounding can be used to achieve approximate solutions (i.e., rather than exact optimal solutions). Randomization incurs no violation of instantaneous constraints; it also preserves the expectation for the violation bound analysis for the long-term constraints, and attains the desired economic properties, both of which will be elaborated later.

3.4 Payment Allocation

Algorithm 4 calculates the payment to each bid in the auction that occurs at the time slot t. If bid n is a winning bid, its payment is as shown in Line 3, consisting of two parts. $\tilde{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t})$ represents the fractional solution as shown in Algorithm 1. $b_{n,t}$ denotes the bidding price of bid n, and $\mathbf{b}_{-n,t}$ denotes the bidding prices of all the other bids.

Algorithm 4 Online Payment Allocation (OPA), $\forall t$

```
1: for $n \in \mathcal{N}$ do
2:   if $\bar{x}_{n,t} == 1$ then
3:     $w_{n,t} = b_{n,t}\tilde{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t}) + \int_{b_{n,t}}^{\varpi_t} \tilde{x}_{n,t}(b, \mathbf{b}_{-n,t})\mathrm{d}b$;
4:   end if
5:   if $\bar{x}_{n,t} == 0$ then
6:     $w_{n,t} = 0$;
7:   end if
8: end for
```

ϖ_t in the integration represents an upper bound, intuitively reflecting the maximum unit payment the service provider can tolerate and make to the bid. If bid n is not a winning bid, then it just receives no payment, as in Line 6.

Insights: Algorithm 4 follows the sufficient and necessary conditions [36] for randomized auctions to be truthful and individual rationality. The reason that we can directly design our payment algorithm like this is that we have ensured $E(\bar{\boldsymbol{x}}_t) = \tilde{\boldsymbol{x}}_t$ in our previous algorithms.

4 Performance Analysis

We define and analyze (i) the regret and the fit, (ii) the competitive ratio, and (iii) the truthfulness and the individual rationality in Sections 4.1∼4.3, respectively, whose detailed proofs are in Sections 4.4∼4.6 correspondingly. We note that the regret and the fit are for the sub-problem $\mathscr{P}_2$; the competitive ratio, the truthfulness, and the individual rationality are for the original holistic problem $\mathscr{P}$.

Notations: For additional notations, we use $\{\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*\}$ to denote the offline optimal integer solution to the problem $\mathscr{P}$. For each t, given $\bar{\boldsymbol{x}}_t$, we use $\{\tilde{\boldsymbol{y}}_t^*, \tilde{\boldsymbol{z}}_t^*\}$ to denote the offline optimal fractional solution to the one-shot slice of the problem $\mathscr{P}_2$ at t.

4.1 Regret and Fit

The regret measures the difference between an optimization problem's objective value incurred by an online algorithm and that incurred by the series of one-shot offline optimums. The fit measures the violation of the constraints. Typically, we expect the regret and the fit to be sub-linear with time. That is, they grow slower than time elapses; in other words, the time-averaged regret and the time-averaged fit diminish as time goes. We formally define regret and fit, respectively, as below and afterwards show that for the problem $\mathscr{P}_2$, our proposed approach indeed leads to sub-linear regret and fit.

Definition 1 (Regret [37])**.** For the optimization problem

$$\begin{array}{ll} \min_{\mathcal{X}_t \in \mathbb{X}} & \sum_{t=1}^{T} f_t(\mathcal{X}_t) \\ \text{s.t.} & h_t(\mathcal{X}_t) \preceq \mathbf{0}, \forall t; \sum_{t=1}^{T} g_t(\mathcal{X}_t) \preceq \mathbf{0}, \end{array}$$

whose one-shot slice at t is

$$\begin{array}{ll} \min_{\mathcal{X}_t \in \mathbb{X}} & f_t(\mathcal{X}_t) \\ \text{s.t.} & h_t(\mathcal{X}_t) \preceq \mathbf{0}; g_t(\mathcal{X}_t) \preceq \mathbf{0}, \end{array}$$

we define the *regret* associated to an online algorithm as

$$Reg = \textstyle\sum_{t=1}^{T} (f_t(\bar{\mathcal{X}}_t) - f_t(\mathcal{X}_t^*)),$$

where $\{\bar{\mathcal{X}}_t, \forall t\}$ represent the solutions returned by this online algorithm and $\{\mathcal{X}_t^*, \forall t\}$ represent the series of the one-shot offline optimal solutions, i.e., for each t, $\mathcal{X}_t^*$ minimizes the aforementioned one-shot slice problem at t.

Definition 2 (Fit [32])**.** For the aforementioned optimization problem, we define the *fit* associated to an online algorithm as

$$Fit = \textstyle\sum_{t=1}^{T} ||[g_t(\bar{\mathcal{X}}_t)]^+||,$$

where $\{\bar{\mathcal{X}}_t, \forall t\}$ represent the solutions returned by this online algorithm, and $[\cdot]^+ \triangleq \max\{\cdot, 0\}$.

The instantaneous constraints are always respected, and therefore we only consider the fit for the long-term constraints. We present the following theorem, where the expectation is due to the randomization introduced in rounding:

Theorem 1. Given $\{\bar{\boldsymbol{x}}_t, \forall t\}$ as the input, the regret and the fit of the problem $\mathscr{P}_2$ incurred by our algorithms are

$$\textstyle\sum_{t=1}^{T} \left(E[\Theta_{-S}^{t}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t)] - \Theta_{-S}^{t}(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*)\right) \leq \mathcal{O}(T^{\varsigma_1}),$$
$$\textstyle\sum_{t=1}^{T} ||[E[\boldsymbol{g}_t(\bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t)]]^+|| \leq \mathcal{O}(T^{\varsigma_2}),$$

where $\varsigma_1, \varsigma_2 \in (0, 1)$.

Proof. See Section 4.4. □

4.2 Competitive Ratio

The competitive ratio measures the multiplicative gap between an optimization problem's objective value incurred by an online algorithm against that incurred by the offline optimum. Typically, we expect such a ratio to be a constant that is independent of time. That is, as time goes, the ratio stays unchanged. The formal definition is as follows.

Definition 3 (Competitive Ratio)**.** For a minimization problem, the competitive ratio ς of an online algorithm satisfies $\mathcal{P}(\bar{\mathcal{X}}) \leq \varsigma \cdot \mathcal{P}(\mathcal{X}^*)$, where $\mathcal{P}(\cdot)$ refers to the objective function; $\bar{\mathcal{X}}$ represents the solution returned by this online algorithm; and $\mathcal{X}^*$ represents the offline optimal solution.

The offline optimal solution refers to the optimal solution to the problem as a whole, instead of breaking the problem into one-shot slices and solving each slice to its optimum individually. Thus, the offline optimum is different from the series of one-shot optimums in Section 4.1, where the latter can better align with existing literature on online learning. Using the offline optimum in the definition of regret can be of independent interest, outside the scope of this paper. We highlight that our algorithms indeed possess a constant competitive ratio via the following theorem.

Theorem 2. For the problem $\mathscr{P}$, the switching cost and the non-switching cost incurred by our algorithms satisfy

$$\textstyle\sum_{t=1}^{T} \Theta_S^t(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{x}}_{t-1}) \leq \Xi \sum_{t=1}^{T} \Theta_{-S}^{t}(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*),$$

and the total cost incurred by our algorithms satisfies

$$\sum_{t=1}^{T}[\Theta_S^t(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{x}}_{t-1}) + \Theta_{-S}^{t}(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*)] \leq \varsigma_2 \sum_{t=1}^{T} \Theta^t(\boldsymbol{x}_t^*, \boldsymbol{y}_t^*, \boldsymbol{z}_t^*),$$

where $\varsigma_2 = \xi(1 + \Xi)$. Based on this, for the problem $\mathscr{P}$, our algorithms lead to

$$E[\textstyle\sum_{t=1}^{T} \Theta^t(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t)] \leq \varsigma \cdot \sum_{t=1}^{T} \Theta^t(\boldsymbol{x}_t^*, \boldsymbol{y}_t^*, \boldsymbol{z}_t^*),$$

where ς is the competitive ratio independent of time.

Proof. See Section 4.5. □

4.3 Truthfulness and Individual Rationality

Intuitively, the utility of a bid is the difference between the payment received from the auctioneer and the true cost of the bid. Based on that, one can define truthfulness, which means every bid maximizes its utility by using its true cost as the bidding price (i.e., no motivation to lie about cost) and individual rationality, which means every bid always has non-negative utility regardless of the auction outcome (i.e., no loss for voluntary participation of the auction). After presenting the formal definitions, we show that our algorithms have indeed achieved the desired economic properties of truthfulness and individual rationality in each auction.

Definition 4 (Utility)**.** The utility of bid n for a randomized auction at the time slot t is

$$\nu_n(b_{n,t}, \mathbf{b}_{-n,t}) = \begin{cases} \omega_{n,t}(b_{n,t}, \mathbf{b}_{-n,t}) - b'_{n,t} E[\bar{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t})], \\ \quad \text{if } \bar{x}_{n,t} = 1, \\ 0, \quad \text{otherwise,} \end{cases}$$

where $b_{n,t}$ is the bidding price; $b'_{n,t}$ is the true cost; $\mathbf{b}_{-n,t}$ represents the bidding prices of all other bids except bid n; $\omega_{n,t}(b_{n,t}, \mathbf{b}_{-n,t})$ denotes the payment, which depends on $b_{n,t}$ and $\mathbf{b}_{-n,t}$; and $\bar{x}_{n,t}$ indicates whether bid n wins in this auction or not, also depending on $b_{n,t}$ and $\mathbf{b}_{-n,t}$. The expectation is because this is for a randomized auction.

Definition 5 (Truthfulness)**.** A randomized auction at t is truthful in expectation if and only if every bid can achieve the maximum utility when it bids its true cost, i.e., $\nu_n(b'_{n,t}, \mathbf{b}_{-i,t}) \geq \nu_n(b_{n,t}, \mathbf{b}_{-n,t}), \forall b_{n,t} \neq b'_{n,t}, \forall n.$

Definition 6 (Individual Rationality)**.** A randomized auction at t is individually rational in expectation if and only if every bid has non-negative utility, i.e., $\nu_n(b_{n,t}, \mathbf{b}_{-n,t}) \geq 0$, $\forall n$.

Theorem 3. *Our algorithms make the auction at each time slot truthful and individually rational.*

Proof. See Section 4.6. Via $E[\bar{x}_{n,t}] = \widetilde{x}_{n,t}$. Also via the necessary and sufficient conditions for a randomized auction to be truthful and individually rational [36]. □

4.4 Proof of Theorem 1

Following Definitions 1∼2, we first prove that the regret and fit in the real domain is sub-linear with time, detailed illustration in Lemma 1. Then, we link the regret/fit in the real domain and integral domain via $E[\bar{\boldsymbol{y}}_t] = \widetilde{\boldsymbol{y}}_t$ and $E[\bar{\boldsymbol{z}}_t] = \widetilde{\boldsymbol{z}}_t$.

Lemma 1. *The regret and fit of $\mathscr{P}_2$ in the real domain incurred by Algorithm 2 grows sub-linearly with time:*

$$\textstyle\sum_{t=1}^T\{\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)\} - \sum_{t=1}^T\{\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*)\} \leq \mathcal{O}(T^{\varsigma_1}),$$
$$\textstyle\sum_{t=1}^T ||[\boldsymbol{g}_t(\widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)]^+|| \leq \mathcal{O}(T^{\varsigma_2}),$$

where $\varsigma_1, \varsigma_2 \in (0,1)$.

Proof. For simplicity, we use $f_t(\tilde{\mathcal{X}}_t)$, $f_t(\tilde{\mathcal{X}}_t^*)$ and $g_t(\tilde{\mathcal{X}}_t)$ to represent $\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)$, $\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*)$ and $\boldsymbol{g}_t(\widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)$, respectively. Therefore, we need to prove $\sum_{t=1}^T(f_t(\tilde{\mathcal{X}}_t) - f_t(\mathcal{X}_t^*)) \leq \mathcal{O}(T^{\varsigma_1})$ and $\sum_{t=1}^T ||[g_t(\tilde{\mathcal{X}}_t)]^+|| \leq \mathcal{O}(T^{\varsigma_2})$.

Before proving it, some commonly adopted and easily met assumptions, which are also widely used in similar settings [37], [38], are introduced as follows: (i) $\forall t, g_t(\tilde{\mathcal{X}}_t)$ is bounded (i.e., $||g_t(\tilde{\mathcal{X}}_t)|| \leq G, \forall \tilde{\mathcal{X}}_t \in \mathbb{X}, \forall t \in \mathcal{T}$, where G is a constant and $\mathbb{X}$ is the convex domain); (ii) The radius of convex domain $\mathbb{X}$ can be bounded by R (i.e., $||\tilde{\mathcal{X}}_i - \tilde{\mathcal{X}}_j|| \leq R, \forall \tilde{\mathcal{X}}_i, \tilde{\mathcal{X}}_j \in \mathbb{X}$, where R is a constant).

For convex function $f_t(\cdot)$, we have $f_t(b) \geq f_t(a) + \nabla f_t^T(b-a)$. After taking $a = \tilde{\mathcal{X}}_t$ and $b = \tilde{\mathcal{X}}_t^*$, we obtain $f_t(\tilde{\mathcal{X}}_t^*) \geq f_t(\tilde{\mathcal{X}}_t) + \nabla f_t^T(\tilde{\mathcal{X}}_t^* - \tilde{\mathcal{X}}_t)$. After rearranging it, we have the following equalities:

$$\begin{aligned} &f_t(\tilde{\mathcal{X}}_t) - f_t(\tilde{\mathcal{X}}_t^*) \leq -\nabla f_t^T(\tilde{\mathcal{X}}_t^* - \tilde{\mathcal{X}}_t) \overset{(a)}{\leq} \mu\beta_1 + \alpha\beta_2 + \\ &\frac{(\beta_3||\tilde{\mathcal{X}}_t^* - \tilde{\mathcal{X}}_{t-1}^*|| + \beta_4(||\tilde{\mathcal{X}}_t - \tilde{\mathcal{X}}_{t-1}^*||^2 - ||\tilde{\mathcal{X}}_t^* - \tilde{\mathcal{X}}_{t+1}^*||^2))}{\alpha} \\ &+ \frac{\beta_5(||\boldsymbol{\omega}_{t+1}||^2 - ||\boldsymbol{\omega}_{t+2}||^2)}{\mu}, \end{aligned} \tag{2}$$

where Inequality (2)a holds since Lemma 6 of the previous work [37] and $\beta_1 \sim \beta_5$ are all constants in this lemma. Then, we take the sum for the equality above from $t = 1$ to $t = T$ and obtain:

$$\begin{aligned} &\textstyle\sum_t f_t(\tilde{\mathcal{X}}_t) - \sum_t f_t(\tilde{\mathcal{X}}_t^*) \overset{(a)}{\leq} T * (\mu\beta_1 + \alpha\beta_2) \\ &+ (\beta_3 * \bar{V}_I + \beta_4 * (||\tilde{\mathcal{X}}_1 - \tilde{\mathcal{X}}_0^*||^2 - ||\tilde{\mathcal{X}}_T^* - \tilde{\mathcal{X}}_{T+1}^*||^2))/\alpha \\ &+ \beta_5 * (||\boldsymbol{\omega}_2||^2 - ||\boldsymbol{\omega}_{T+2}||^2)/\mu \\ &\overset{(b)}{\leq} T(\mu\beta_1 + \alpha\beta_2) + \frac{\beta_3\bar{V}_I + \beta_4||\tilde{\mathcal{X}}_1 - \tilde{\mathcal{X}}_0^*||^2}{\alpha} + \frac{\beta_5(||\boldsymbol{\omega}_2||^2)}{\mu} \\ &\overset{(c)}{\leq} T(\mu\beta_1 + \alpha\beta_2) + (\beta_3\bar{V}_I + \beta_4R^2)/\alpha + \beta_5\mu G^2 \\ &\overset{(d)}{=} \mathcal{O}(\max\{T^{\frac{1}{\alpha_1}}, T^{\frac{\alpha_1-1}{\alpha_1}}\}) \triangleq \mathcal{O}(T^{\varsigma_1}), \end{aligned} \tag{3}$$

where Inequality (3)a holds since $\bar{V}_I = \sum_{t\in\mathcal{T}} ||\tilde{\mathcal{X}}_t^* - \tilde{\mathcal{X}}_{t-1}^*||$ which is the accumulated variation of the per-slot minimizers $\tilde{\mathcal{X}}_t^*$. (3)b holds since $||\tilde{\mathcal{X}}_T^* - \tilde{\mathcal{X}}_{T+1}^*||^2 \geq 0$ and $||\boldsymbol{\omega}_{T+2}||^2 \geq 0$; (3)c holds since the bounded domain of Assumption (ii) and $||\boldsymbol{\omega}_2||^2 \leq (\mu G)^2$ of Assumption (i); (3)d holds since we take $\mu = \alpha = \mathcal{O}(T^{-\frac{1}{\alpha_1}})$, $\alpha_1 > 2$.

According to the definition of ω in Line 1 of Algorithm 2, we have $\boldsymbol{\omega}_t = [\boldsymbol{\omega}_{t-1} + \mu g_{t-1}(\tilde{\mathcal{X}}_{t-1})]^+ \geq \boldsymbol{\omega}_{t-1} + \mu g_{t-1}(\tilde{\mathcal{X}}_{t-1}) \geq \boldsymbol{\omega}_{t-2} + \mu g_t(\tilde{\mathcal{X}}_{t-2}) + \mu g_{t-1}(\tilde{\mathcal{X}}_t) \geq ... \geq \mu \sum_t g_t(\tilde{\mathcal{X}}_t) + \boldsymbol{\omega}_1 = \mu \sum_t g_t(\tilde{\mathcal{X}}_t)$, obtaining:

$$\begin{aligned} &\|[\textstyle\sum_t \boldsymbol{g}_t(\tilde{\mathcal{X}}_t)]^+\| \overset{(a)}{\leq} \|\sum_t \boldsymbol{g}_t(\tilde{\mathcal{X}}_t)\| \overset{(b)}{\leq} \|\boldsymbol{\omega}_{T+1}\|/\mu \\ &\overset{(c)}{=} \beta_6/(\alpha\mu) + \beta_7/\mu + \beta_8 \overset{(d)}{=} \mathcal{O}(T^{\frac{2}{\alpha_1}}) = \mathcal{O}(T^{\varsigma_2}), \end{aligned} \tag{4}$$

where $\beta_6 \sim \beta_8$ are constants. Equality (4)a holds since the nature of $[\cdot]^+$; (4)b holds since the above equalities. (4)c holds since Lemma 2 of previous work [1] and $\beta_6 \sim \beta_8$ are all constants from this lemma. (4)d holds since we take $\mu = \alpha = \mathcal{O}(T^{-\frac{1}{\alpha_2}})$. We have $\varsigma_2 \in (0,1)$, since $\alpha_1 > 2$. □

Based on Lemma 1, we have the derivation:

$$\begin{aligned} &\textstyle\sum_{t=1}^T\{E[\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t)]\} - \sum_{t=1}^T\{\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*)\} \\ &\overset{(a)}{=} \textstyle\sum_{t=1}^T\{\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)\} - \sum_{t=1}^T\{\Theta_{-S}^t(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t^*, \widetilde{\boldsymbol{z}}_t^*)\} \\ &\overset{(b)}{\leq} \mathcal{O}(T^{\varsigma_1}), \end{aligned} \tag{5}$$

where (5)a holds since $E[\bar{\boldsymbol{y}}_t] = \widetilde{\boldsymbol{y}}_t$ and $E[\bar{\boldsymbol{z}}_t] = \widetilde{\boldsymbol{z}}_t$; (5)b holds since Lemma 1;

As for the fit incurred by our algorithm for $\mathscr{P}_2$, we have the derivation:

$$\begin{aligned} &\textstyle\sum_{t=1}^T ||[E[\boldsymbol{g}_t(\bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t)]]^+|| \\ &\overset{(a)}{=} \textstyle\sum_{t=1}^T ||[\boldsymbol{g}_t(\widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)]^+|| \overset{(b)}{\leq} \mathcal{O}(T^{\varsigma_2}), \varsigma_2 \in (0,1), \end{aligned} \tag{6}$$

where (6)a holds since $E[\bar{\boldsymbol{y}}_t] = \widetilde{\boldsymbol{y}}_t$ and $E[\bar{\boldsymbol{z}}_t] = \widetilde{\boldsymbol{z}}_t$; (6)b holds since Lemma 1.

4.5 Proof of Theorem 2

We first connect the switching cost term $\Theta_{-S}^u(\bar{\boldsymbol{x}}_u, \widetilde{\boldsymbol{y}}_u^*, \widetilde{\boldsymbol{z}}_u^*)$, abbreviated as $\Theta_{-S}^u(\hat{\boldsymbol{X}}_u)$, to the non-switching cost term $\Theta_S^u(\bar{\boldsymbol{x}}_u, \bar{\boldsymbol{x}}_{u-1})$, abbreviated as $\Theta_{-S}^u(\hat{\boldsymbol{X}}_u)$, via the control parameter η. We then connect $\Theta_{-S}^u(\hat{\boldsymbol{X}}_u)$ to its optimum $\Theta_{-S}^u(\boldsymbol{x}_u^*, \boldsymbol{y}_u^*, \boldsymbol{z}_u^*)$, abbreviated as $\Theta_{-S}^u(\boldsymbol{X}_u^*)$.

Here, we link the switching cost and non-switching cost. We use t_l', $\forall 1 \leq l \leq l'$ to denote the timestamps when the switch operations occur, where l' is the total number of the switch operations over the time horizon under consideration. The switching cost incurred at t_l' is $\Theta_S^{t_l'}(\hat{\boldsymbol{X}}_{t_l'})$. The non-switching cost accumulated during the period $[t_l', t_{l+1}' - 1]$ is at least $\frac{1}{\eta}$ times the switching cost at t_l', denoted by

$$\frac{1}{\eta}\Theta_S^{t_l'}(\hat{\boldsymbol{X}}_{t_l'}) \leq \sum_{u=t_l'}^{t_{l+1}'-1} \Theta_{-S}^u(\hat{\boldsymbol{X}}_u), \quad \forall 1 \leq l \leq l'. \tag{7}$$

Furthermore, when the switching operation occurs in time slot t_l', the ratio of potential switching cost to potential non-switching cost can be bounded as follows:

$$\frac{\Theta_S^{t_l'}(\hat{\boldsymbol{X}}_{t_l'})}{\Theta_{-S}^{t_l'}(\hat{\boldsymbol{X}}_{t_l'})} \overset{(a)}{\leq} \frac{\sum_{n=1}^N l_n}{\min_n\{b_{n,t_l'}\}} \triangleq \vartheta, \quad \forall 1 \leq l \leq l', \tag{8}$$

where Inequality (8)a holds since the numerator of the middle term (i.e., $\sum_{n=1}^{N} l_n$) is the upper bound of the switching cost in each time slot, while the denominator (i.e., $\min_n\{b_{n,t'_l}\}$) represents the lower bound of the non-switching cost. Therefore, for all $1 \leq t \leq T$, we have the following derivation equations:

$$\sum_{u=1}^{t} \Theta_S^u(\hat{\boldsymbol{X}}_u) = \sum_{1\leq l\leq l'} \sum_{u=t'_l}^{t'_{l+1}-1} \Theta_S^u(\hat{\boldsymbol{X}}_u) + \sum_{u=t'_{l'}+1}^{t} \Theta_S^u(\hat{\boldsymbol{X}}_u)$$

$$= \sum_{1\leq l<l'} \{\Theta_S^{t'_l}(\hat{\boldsymbol{X}}_{t'_l}) + \sum_{u=t'_l+1}^{t'_{l+1}-1} 0\} + \Theta_S^{t'_{l'}}(\hat{\boldsymbol{X}}_{t'_l}) + \sum_{u=t'_{l'}+1}^{t} 0$$

$$\overset{(a)}{\leq} \sum_{1\leq l<l'} \{\eta \sum_{u=t'_l}^{t'_{l+1}-1} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u)\} + \vartheta \Theta^{t'_{l'}}_{-S}(\hat{\boldsymbol{X}}_{t'_{l'}}) + \vartheta \sum_{u=t'_{l'}+1}^{t} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u)$$

$$= \eta \sum_{u=1}^{t'_{l'}-1} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u) + \vartheta \sum_{u=t'_{l'}}^{t} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u)$$

$$\leq \max\{\eta, \vartheta\} \sum\nolimits_{u=1}^{t} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u) \triangleq \Xi \sum\nolimits_{u=1}^{t} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u), \quad (9)$$

where Ξ is a constant. And Inequality (9)a holds since the Inequalities (7) & (8) and the extra added term (i.e., $\vartheta \sum_{u=t'_{l'}+1}^{t} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u)$) is non-negative.

Then, we take $\xi \triangleq \max_u \frac{\max_{\hat{\boldsymbol{X}}_u} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u)}{\min_{\hat{\boldsymbol{X}}_u} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u)}$, and have

$$\Theta^u_{-S}(\hat{\boldsymbol{X}}_u) \leq \xi \Theta^u_{-S}(\boldsymbol{X}^*_u), \forall u \leq T,$$

where we link the non-switching cost corresponding to solution $\{\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}^*_t, \widetilde{\boldsymbol{z}}^*_t\}$ and offline optimum $\{\boldsymbol{x}^*_t, \boldsymbol{y}^*_t, \boldsymbol{z}^*_t\}$.

After taking $u = 1$ to $u = t$, $\forall t \leq T$, we have the following inequality:

$$\sum\nolimits_{u=1}^{t} \Theta^u_{-S}(\hat{\boldsymbol{X}}_u) \leq \xi \sum\nolimits_{u=1}^{t} \Theta^u_{-S}(\boldsymbol{X}^*_u) \leq$$
$$\xi \sum\nolimits_{u=1}^{t} \{\Theta^u_S(\boldsymbol{X}^*_u) + \Theta^u_{-S}(\boldsymbol{X}^*_u)\} \leq \xi \sum\nolimits_{u=1}^{t} \Theta^u(\boldsymbol{X}^*_u),$$

where we link the non-switching cost and the total cost.

Therefore, the overall cost can be bounded by

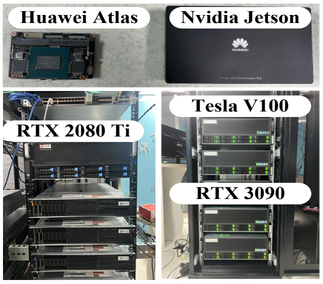
$$\sum_{t=1}^{T} [\Theta^t_S(\hat{\boldsymbol{X}}_t) + \Theta^t_{-S}(\hat{\boldsymbol{X}}_t)] \leq (1+\Xi) \sum_{t=1}^{T} \Theta^t_{-S}(\hat{\boldsymbol{X}}_t)$$

$$\leq \xi(1+\Xi) \sum_{t=1}^{T} \Theta^t(\boldsymbol{X}^*_t) \triangleq \varsigma_2 \sum\nolimits_{t=1}^{T} \Theta^t(\boldsymbol{X}^*_t) = \varsigma_2 \mathcal{P}^*. \quad (10)$$

Then, for Theorem 2, we have the following derivation:

$$E[\mathcal{P}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})] = E[\sum\nolimits_{t=1}^{T} \{\Theta^t_S(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{x}}_{t-1}) + \Theta^t_{-S}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{y}}_t, \bar{\boldsymbol{z}}_t)\}]$$
$$\overset{(a)}{\leq} \sum\nolimits_{t=1}^{T} \{\Theta^t_S(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{x}}_{t-1}) + \Theta^t_{-S}(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}_t, \widetilde{\boldsymbol{z}}_t)\}$$
$$\overset{(b)}{\leq} \sum\nolimits_{t=1}^{T} \{\Theta^t_S(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{x}}_{t-1}) + \Theta^t_{-S}(\bar{\boldsymbol{x}}_t, \widetilde{\boldsymbol{y}}^*_t, \widetilde{\boldsymbol{z}}^*_t) + \mathcal{O}(T^{\varsigma_1})\}$$
$$\overset{(c)}{\leq} \varsigma_2 \mathcal{P}^* + \mathcal{O}(T^{\varsigma_1})\} \triangleq \varsigma \mathcal{P}^*, \quad (11)$$

where (11)a holds since $E[\bar{\boldsymbol{y}}_t] = \widetilde{\boldsymbol{y}}_t$ and $E[\bar{\boldsymbol{z}}_t] = \widetilde{\boldsymbol{z}}_t$; (11)b holds since Equality 5(b) in Lemma 1; (11)c holds since Equality (10). Please note that given T, $\mathcal{O}(T^{\varsigma_1})$ is a constant.



Device	Latency
Edge Clients	
Huawei	311 ms
Jetson	191 ms
Edge Servers	
Tesla V100	24 ms
RTX 3090	25 ms
RTX 2080Ti	39 ms

(a) Real Devices (b) Inference Latency

Fig. 6: Testbed Setup and Measurement

4.6 Proof of Theorem 3

(1) Proof of *truthfulness* for our randomized auction. We first introduce the following proposition [36]:

Proposition 1. A randomized auction is *Truthfulness* in expectation if and only if (i) $E[\bar{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t})]$ is monotonically non-increasing at $b'_{n,t}$ for any bidder n, when all inputs except for n remain unchanged; (ii) The upper bound of the integral must be less than infinity, i.e., $\int_{b_{n,t}}^{\infty} E[\bar{x}_{n,t}(b, \mathbf{b}_{-n,t})]\mathrm{d}b < \infty$;

From Proposition 1, we begin the following two proofs: (i) For the problem $\mathscr{P}_1$, the objective function of each time slot is $\Theta^t_{-S}(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)$. For each bidder n, given the inputs except for $x_{n,t}$ and $b_{n,t}$, we convert this function as $\Theta^t_{-S,n}(x_{n,t}, b_{n,t}|\mathbf{b}_{-n,t})$ with variables $x_{n,t}$ and $b_{n,t}$. Assuming that $\widetilde{x}_{n,t}$ and $\widetilde{x}^1_{n,t}$ are the optimal solutions with the bidding prices of $b_{n,t}$ and $b^1_{n,t}$, we have the equalities:

$$\Theta^t_{-S,n}(\widetilde{x}_{n,t}, b_{n,t}|\mathbf{b}_{-n,t}) \leq \Theta^t_{-S,n}(\widetilde{x}^1_{n,t}, b_{n,t}|\mathbf{b}_{-n,t}),$$
$$\Theta^t_{-S,n}(\widetilde{x}^1_{n,t}, b^1_{n,t}|\mathbf{b}_{-n,t}) \leq \Theta^t_{-S,n}(\widetilde{x}_{n,t}, b^1_{n,t}|\mathbf{b}_{-n,t}).$$

Add two inequalities together, reorganize them, and we have: $(b_{n,t}-b^1_{n,t})*\widetilde{x}_{n,t} \leq (b_{n,t}-b^1_{n,t})*\widetilde{x}^1_{n,t}$. Take $b_{n,t}-b^1_{n,t} > 0$, and we have $\widetilde{x}_{n,t} \leq \widetilde{x}^1_{n,t}$. Since $E[\bar{x}_{n,t}] = \widetilde{x}_{n,t}$ and $E[\bar{x}^1_{n,t}] = \widetilde{x}^1_{n,t}$, we obtain $E[\bar{x}_{n,t}] \overset{(a)}{\leq} E[\bar{x}^1_{n,t}]$.

From Proposition 1, we make the other proof (ii):

$$\int_{b_{n,t}}^{\infty} E[\bar{x}_{n,t}(b, \mathbf{b}_{-n,t})]\mathrm{d}b = \int_{b_{n,t}}^{\varpi_t} E[\bar{x}_{n,t}(b, \mathbf{b}_{-n,t})]\mathrm{d}b \leq \varpi_t < \infty.$$

As for $\mathscr{P}_2$, we have a similar conclusion.

(2) Proof of *individual rationality* for our randomized auction. For the utility of each winning bidder n, we have:

$$\nu_n(b_{n,t}, \mathbf{b}_{-n,t}) = w_{n,t}(b_{n,t}, \mathbf{b}_{-n,t}) - b'_{n,t} E[\bar{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t})]$$
$$= b_{n,t}\widetilde{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t}) + \int_{b_{n,t}}^{\varpi_t} \widetilde{x}_{n,t}(b, \mathbf{b}_{-n,t})\mathrm{d}b$$
$$- b'_{n,t} E[\bar{x}_{n,t}(b_{n,t}, \mathbf{b}_{-n,t})] \overset{(a)}{\geq} \int_{b_{n,t}}^{\varpi_t} \widetilde{x}_{n,t}(b, \mathbf{b}_{-n,t})\mathrm{d}b \geq 0, \quad (12)$$

where Inequality(12)a holds since $E[\bar{x}_{n,t}] = \widetilde{x}_{n,t}$ and $b_{n,t} \geq b'_{n,t}$.

5 Experimental Evaluations

5.1 Evaluation Setup

We evaluate the performance of our proposed approach in practice, and compare it to multiple baseline and state-of-the-art approaches. For the inputs to all these algorithms,

TABLE 2: Parameter Values

Para.	Value
$b_{n,t}$	Bidding price in $\$4 \sim \$18$ from AWS [39]
l_n	Switching cost in $10 \sim 100$s [40], [41]
c_n	Computing capacity in $4 \sim 128$ cores [39]
q_n	Queue capacity in $204 \sim 8100$ [42]
τ_t	# of passengers [21], [43]–[45], # $50 \sim 1250$ queries/passenger [46]
$d_{n,t}$	Comm. cost per query in $0.1 \sim 0.9$s [47], [48]
$a_{n,m,t}$	Error rate in $0.1 \sim 0.3$ [46]
r_m	Resource requirement in $1 \sim 20$ cores [49]
$p_{n,m}$	Processing speed in $105 \sim 2447$ from testbed and [50]
$e_{n,m,t}$	Model transmission cost in $50 \sim 850$s [1]
$\delta_{n,m,t}$	Traces on whether to send models to edges [51]

we use our lab testbed measurement results and data from other existing works, as in Table 2. The details are as follows.

Testbed-Based Measurement: We construct a real-world testbed consisting of multiple hardware components, including the GeForce RTX 2080Ti, the GeForce RTX 3090, the Nvidia Tesla V100, the Huawei Atlas 200DK, and the Nvidia Jetson NX, as shown in Fig. 6(a). On this testbed, we measure the inference performance of different real-world models (on computer vision and natural language processing): YOLOv4 [52], DeepLabv3 [53], BERT [54], VGG16 [55], and Inceptionv3 [56]. For example, the results of YOLOv4 are displayed in Fig. 6(b). These will be used as the inputs to our subsequent experimental evaluations.

Edge Devices and Bidding: The bidding prices come from the Amazon EC2 platform [39], ranging in $\$4 \sim \$18$. The server's switching cost (e.g., re-connecting time, re-preparing resource time) is set to 10∼100s based on existing statistics [40], [41]. Each bidder's available computing capacity is set to 4∼128 processor cores [1]. The queue capacity is related with the edge device's memory, which is set to 204∼8100 [42], [57]. The processing speed ranges in 105∼2447 requests per time slot, which comes from our testbed measurement results and an existing study [50]. We consider up to 1200 bidders in the system.

Inference Workload: The inference workloads at each edge are obtained from the dynamic passenger counts of the 268 London underground stations [43], which was collected in 15-minute intervals over a four-day period in November 2016, resulting in a total of about 300 time slots. This is our default dataset, denoted by *London*. From this dataset, we set the duration of each single time slot to 15 minutes [1], also aligned with recent studies [1], [11], [46], [58] and datasets [59]. To demonstrate the robustness and applicability of our algorithms across different workloads, we also explore multiple other datasets:

- *IRSMSeting* [21]: Based on the setup in the IRSM study. 5∼30 users generated per time slot.
- *MTA* [45]: Passenger volumes from the Metropolitan Transportation Authority (MTA), including records from subway, MTA Staten Island Railway (SIR), and Roosevelt Island tram systems.
- *Telecom* [44]: Shanghai Telecom dataset, containing over 7.2 million records collected over six months and involving 9,481 mobile devices accessing the Internet via 3,233 base stations.

For *London*, we consider that each passenger submits 50∼1250 inference queries [46] at each time slot; for the other datasets, we consider that each user or device submits one inference query at each time slot. Then, at each time slot, we sum up the total number of inference queries received and use this as the inference workload for the edge AI service. The communication cost for sending one query ranges in 0.1∼0.9s [1], [42].

Machine Learning Settings: We set the models' inference error rates and resource requirements to 0.1∼0.3 and 1∼20 cores, respectively [1], [46]. Usually, models with low error rates have high resource requirements. We set the model transmission time to 50∼850s [1]. We obtain the traces [51] for $\delta_{n,m,t}$ regarding whether to send models as time goes. We consider up to more than a dozen of models available for conducting inference in the system.

Algorithm Parameters: The step sizes in Algorithm 2 are set as $\lambda = \mu = 300^{-\frac{1}{3}}$. η in Algorithm 1 is set to $1/2$ [60].

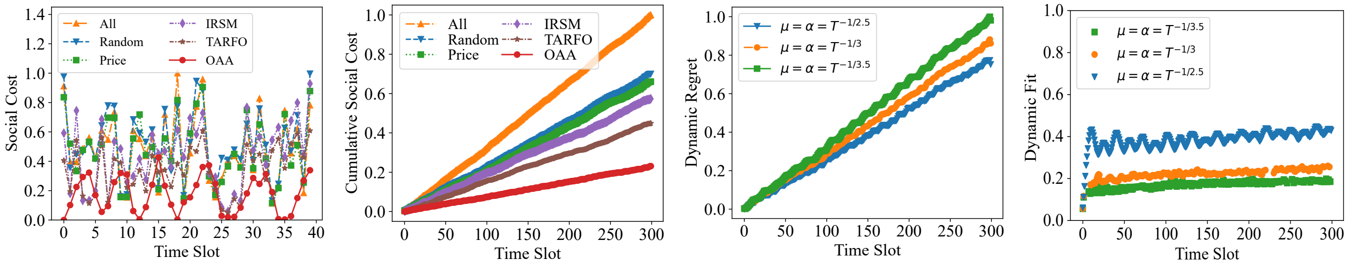
Algorithms for Comparison: We implement our proposed approach *OAA* in Python, using CVXPY [34] for the online learning component. We also implement multiple alternative algorithms for comparison. Our implementations include 900+ lines of Python codes in total:

- *OAA*: Our proposed approach;
- *Random*: Winning bids are chosen randomly;
- *All*: All bids are chosen as the winning bids;
- *Price*: Bids that have the highest cost-effectiveness are chosen as the winning bids;
- *IRSM* [21]: A state-of-the-art approach that maximizes the total social welfare of all edge servers and devices to select the winning bids;
- *TARFO* [20]: A state-of-the-art approach that minimizes the completion time of all the tasks via strategic resource allocation.

5.2 Evaluation Results

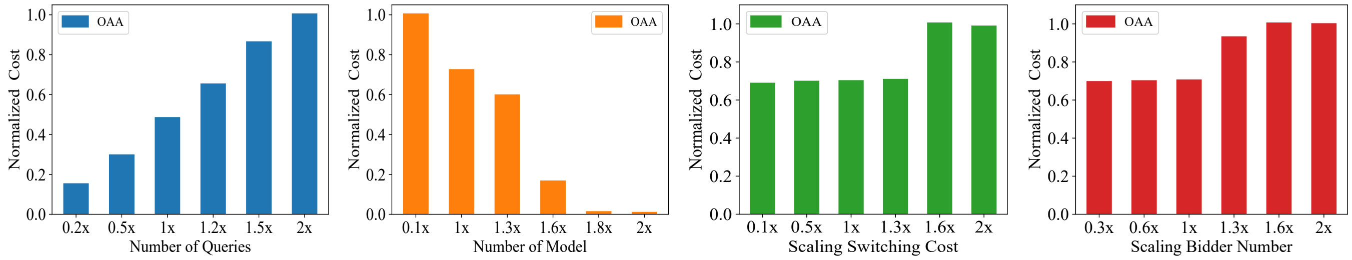
Social Cost: Fig. 7(a) and Fig. 7(b) show the normalized real-time social cost and the normalized cumulative social cost, respectively, incurred by different algorithms. Our approach *OAA* consistently outperforms others, leading to an average cost reduction of about 50%. Compared to the heuristics, i.e., *All*, *Random*, and *Price*, *OAA* reduces the social cost by about 65% on average. Compared to the state-of-the-arts, i.e., *IRSM* and *TARFO*, *OAA* reduces the social cost by about 40% on average. The fluctuation of the real-time cost is due to the system dynamics, i.e., the inputs such as the communication cost, the inference workload, the model error rates, and the bidding prices to each time slot are varying as time goes. The cumulative social cost of *OAA* grows more slowly than others. IRSM and TARFO are for edge computing tasks but do not consider the specific characteristics of AI inference, such as model error rates, while overlooking the switching cost associated with edge devices. As a result, these methods could lead to higher error rates and also increase the social cost. We further calculate the practical competitive ratio of *OAA* and find it to be 1.6∼3.9, with the majority in 1.6∼3, which is good and fully acceptable in practice.

Regret and Fit: Fig. 7(c) and Fig. 7(d) visualize the regret and the fit for our proposed approach, respectively, under different step sizes. The growth of the regret and that of the fit are both sub-linear, which is consistent with our theoretical analysis. We can observe the trade-off between



(a) Real-Time Cost (b) Cumulative Cost (c) Regret (d) Fit

Fig. 7: Social Cost of Different Approaches and Performance of Online Learning



(a) Impact of Workloads (b) Impact of Model Number (c) Impact of Switching Cost (d) Impact of Bidder Number

Fig. 8: Social Cost Impacted by Different Factors

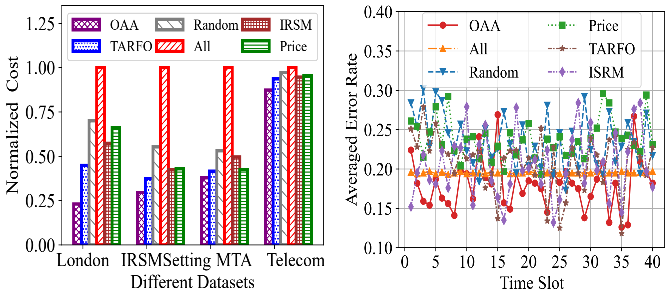


Fig. 9: (a) Cost on Different Datasets; (b) Error Rates

regret and fit: a step size that reduces the regret inevitably increases the fit, which is also aligned with our theoretical analysis. As the step size goes from $T^{-1/2.5}$ to $T^{-1/3}$, the fit decreases by 15%∼25% and the regret grows by 10%∼15%.

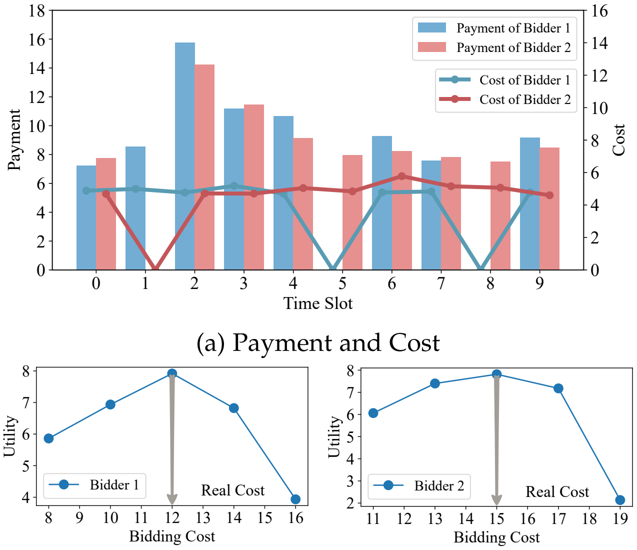
Scalability: Fig. 8 exhibits how the normalized total social cost of our proposed approach is influenced by different factors. In Fig. 8(a), the social cost increases with the growth of the workloads. To process more inference queries, the system needs to recruit more edge devices in each auction, which incurs more social cost. Specifically, as the query workload increases by 0.2x∼0.4x, the social cost increases by 16%∼66%, and the rate of increase gradually slows down. In Fig. 8(b), the social cost decreases with the growth of the number of models. In this figure, as we add new models to the pool of candidate models, the social cost decreases. Note that the capacities of the edge devices do not change. A larger and more heterogeneous pool means more available choices of models with the cost versus accuracy trade-off, thereby resulting in control decisions that achieve lower optimization objective values. Besides, as the number of the models available goes to more than a dozen, e.g., 1.6x∼2x, the social cost can be significantly reduced by more than 80% compared to the case that only has fewer than three models, e.g., 0.1x. Fig. 8(c) depicts the impact on the social cost incurred by the switching cost. The social cost increases with the growth of the switching cost. Fig. 8(d) shows the social cost under varying numbers of bidders. A larger number of bidders can lead to the growth in the social cost. The number of cost-effective edge devices increases with the total number of edge devices, and all of them could have the potential to win the auction.

Fig. 9(a) further investigates how the dynamic workload impacts the social cost. *OAA* consistently achieves the best social cost. Compared to other algorithms, *OAA* reduces the social cost by 63%, 38%, 27%, and 10% on average upon *London*, *IRSMSetting*, *MTA* and *Telecom* datasets, respectively.

Error Rates: Fig. 9(b) visualizes the averaged "error rate" of all the models selected and deployed processing all the inference queries seen in the system over the entire time horizon. The error rate is actually a term already contained in our optimization objective. As shown, the error rate of our approach is 8%∼24% less than the error rates of others.

Individual Rationality and Truthfulness: Fig. 10 demonstrates the economic properties of our approach for two randomly selected bidders. Fig. 10(a) shows the payment and the cost for bidder 1 and bidder 2. The payment calculated by our approach at each time slot is higher than the cost, implying individual rationality; the bidders who lose in an auction receive zero payment. Fig. 10(b) and Fig. 10(c) display the utility functions for bidder 1 and bidder 2, where the maximum benefits are only achieved when bidding at true cost, implying truthfulness.

Running Time: Fig. 11 displays the running time of our proposed approach *OAA*, including its different algorithmic components of online learning (OLA) and randomized rounding (RRA). We run our algorithm codes at different scales on a commercial desktop server with 16 2.5 GHz Intel Core i7 CPUs and 32 GB memory. The results show that our approach consistently finishes in tens of milliseconds.



(a) Payment and Cost

(b) Utility of Bidder 1

(c) Utility of Bidder 2

Fig. 10: Individual Rationality and Truthfulness

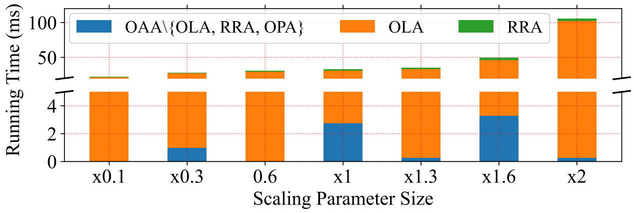


Fig. 11: Running Time

Our OPA algorithm averages around 10 seconds, which is not shown in this figure. As OPA does not have strict time constraints, it only needs to be completed before the end of the time slot (e.g., it can run while conducting inference). Overall, the running time of our approach is significantly less than the length of a single time slot.

6 Related Work

We summarize representative previous work in two groups, and highlight their insufficiency compared to this paper.

6.1 Edge AI Inference Optimization

Liu *et al.* [17] proposed an adaptive Deep Neural Network (DNN) inference framework for mobile edge computing which selects exit and partition points in an online manner. Sun *et al.* [13] developed an online adaptive selection algorithm for the long-term impact of the batches on the inference results. Huang *et al.* [18] used Multi-Exit Deep Neural Networks (ME-DNNs) and a distributed offloading mechanism to reduce latency in edge computing. Jin *et al.* [1] addressed the dynamic volatility of the edge environments via online learning. Miao *et al.* [19] showed a load-balancing algorithm for splitting DNN with Directed Acyclic Graph (DAG) structures to accelerate DNN inference. Wen *et al.* [61] formulated a ISCC problem for optimizing the inference accuracy under the constraints of latency, energy, and transmission, which was solved by the optimal ISCC scheme.

6.2 Edge Incentive Mechanisms

Wang *et al.* [20] proposed a resource allocation algorithm that enforces truthfulness, flexible task offloading, and locality constraints. Li *et al.* [21] proposed a task offloading strategy at edge for dynamic tasks and energy restrictions, improving social welfare. Yuan *et al.* [11] proposed an incentive mechanism for edge federated learning, considering data volume and user dynamism, with performance guarantees. Wang *et al.* [23] devised an incentive mechanism to balance the profit of resource providers with the Quality of Experience of mobile devices. Chen *et al.* [12] incentivized user devices to provide relay services for computation offloading via a two-stage auction model, maximizing utility and considering node authenticity and users' rationality. Wang *et al.* [22] incentivized electric vehicles to discharge energy to support edge computing demand response.

6.3 Research Gap

Table 3 contrasts these prior research with our work in multiple dimensions. Overall, the aforementioned first group of works either assume resources are abundant and readily available, or overlook the resource usage. They do not consider the setting where existing resources are insufficient to meet the varying demand and new resources need to be dynamically recruited and released; they could also incur high operational cost, especially when paying for edge resources in a large scale at a fixed per-unit price, making it impractical for many service providers. The aforementioned second group of works typically have not targeted edge AI inference, or its unique features and challenges as we have identified in this paper, which would lead to degraded inference accuracy and poor service quality if such existing literature and methods were directly applied to the edge AI inference service. This paper bridges all these gaps.

7 Discussions

Bidders' Unstable Connections: Our system runs in a time-slotted manner, where each time slot has one and only one auction. For each single auction, generally, it is assumed that any edge device (i.e., bidder) that participates in the auction stays connected and committed to this auction. This assumption aligns with many existing research on auctions related to cloud and edge computing systems. In a "weird" case where, for example, a bidder submits a bid and leaves the system before the outcome of the current auction, we could strategically exclude this bidder from subsequent auctions due to credibility and reliability issue of this bidder.

Frequent Quitting and Repeated Bidding: Our current work allows arbitrary quitting and repeated bidding *across* auctions. Any bidder can join one auction and does not join another auction, or vice versa. We do not penalize any bidder for this behavior, because each bidder should feel free and volunteer to join or not to join any of the auctions. If a bid does not join or is not selected as a winning bid for some auctions, and then if this bid wins in a later auction, this may incur the switching cost; yet, our current system modeling has already considered such switching cost as part of the social cost that is being minimized.

Applicability of Deep Reinforcement Learning: Deep Reinforcement Learning (DRL) could be relevant, but we do

TABLE 3: Comparing Previous Work to Our Work

Ref.	**Problem Space**				**Solution Space**		**Evaluation**	**Year**
	Edge Incentive	Inference Task	Online Scenario	Long-term Constraints	Approximate/ Competitive Ratio	Economic Properties	TestBed Prototype	
[17]		✓	✓		✓		✓	2024
[13]		✓	✓	✓			✓	2023
[18]		✓	✓	✓			✓	2021
[1]		✓	✓	✓	✓			2020
[19]		✓			✓		✓	2020
[20]	✓				✓	✓	✓	2024
[21]	✓		✓	✓	✓	✓		2023
[22]	✓		✓		✓	✓		2023
[11]	✓		✓	✓	✓	✓		2022
[23]	✓		✓		✓	✓		2022
[12]	✓		✓			✓		2021
Ours	✓	✓	✓	✓	✓	✓	✓	2024

not adopt DRL in this work. First, DRL algorithms often target problems and settings that have no explicit models. Our problem is well-defined with a clear mathematical formulation, and thus, there seems no need to bother DRL. Second, DRL algorithms often lack rigorously-provable guarantees, making it difficult to predict their worst-case performance. In contrast, in this work, our proposed approach is proven to come with multiple performance guarantees. Third, DRL algorithms typically require a large amount of training data and time and lengthy cold-start or convergence overhead. In contrast, our online algorithms make control decisions directly on the fly without the need of "training".

8 Conclusion

Our work in this paper enhances the edge AI service by enabling it to dynamically and continuously recruit resources on a wider range of edge devices and use such resources to provision the inference service at the minimum cost to the end users. We formulate the repeated auctions, and propose a combination of novel online algorithms to jointly control winning-bid selection, model deployment, inference query dispatch, and payment allocation, while overcoming multiple non-trivial challenges. We rigorously analyze our algorithmic approach in terms of multiple performance metrics, and conduct solid experiments to validate the superiority of our design in practice, including comparison to existing baseline and advanced methods. While the focus of this current paper is on inference, we also plan to investigate edge AI training with resource trading in future work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62441225, the Jiangsu Collaborative Innovation Center of Novel Software Technology and Industrialization, and the U.S. National Science Foundation under Grants CNS-2047719 and CNS-2225949. Specifically, thanks to Zhixin Zhang, Runze Ma, Tina, and Radin for their contribution.

References

[1] Y. Jin, L. Jiao, Z. Qian, S. Zhang, N. Chen, S. Lu, and X. Wang, "Provisioning edge inference as a service via online learning," in *IEEE SECON*, 2020.

[2] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Empowering intelligence to the edge of network," *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1778–1837, 2021.

[3] Y. Yang, L. Feng, Y. Sun, Y. Li, W. Li, and M. A. Imran, "Multi-cluster cooperative offloading for vr task: A marl approach with graph embedding," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2024.

[4] "Smart industrial iot empowered crowd sensing for safety monitoring in coal mine," *Digital Communications and Networks*, vol. 9, no. 2, pp. 296–305, 2023.

[5] L. Chen, Y. Zhang, B. Tian, Y. Ai, D. Cao, and F.-Y. Wang, "Parallel driving os: A ubiquitous operating system for autonomous driving in cpss," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 886–895, 2022.

[6] "Building an Edge Strategy: Cost Factors," https://developer.nvidia.com/blog/building-an-edge-strategy-cost-factors/, 2022.

[7] "Amazon SageMaker Edge Manager Simplifies Operating Machine Learning Models on Edge Devices," https://aws.amazon.com/cn/blogs/aws/amazon-sagemaker-edge-manager-simplifies-operating-machine\learning-models-on-edge-devices/, 2020.

[8] "JD Cloud Wireless," https://jdbox.jdcloud.com/, 2024.

[9] "Synology Inc." https://www.synology.cn/, 2024.

[10] "Desktop Computers," https://www.dell.com/en-us/shop/desktop-computers/sc/desktops, 2024.

[11] Y. Yuan, L. Jiao, K. Zhu, and L. Zhang, "Incentivizing federated learning under long-term energy constraint via online randomized auctions," *IEEE Transactions on Wireless Communications*, vol. 21, no. 7, pp. 5129–5144, 2022.

[12] L. Chen, J. Wu, X. Zhang, and G. Zhou, "Tarco: Two-stage auction for d2d relay aided computation resource allocation in hetnet," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 286–299, 2021.

[13] H. Sun, X. Chen, Z. Qian, Z. Li, N. Chen, T. Cao, S. Xu, and Y. Zhou, "Birp: Batch-aware inference workload redistribution and parallel scheme for edge collaboration," in *IEEE ICPP*, 2023.

[14] M. Xiao, B. An, J. Wang, G. Gao, S. Zhang, and J. Wu, "Cmab-based reverse auction for unknown worker recruitment in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3502–3518, 2022.

[15] N. Nisan and A. Ronen, "Computationally feasible vcg mechanisms," *Journal of Artificial Intelligence Research*, vol. 29, pp. 19–47, 2007.

[16] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *IEEE INFOCOM*, 2014.

[17] Z. Liu, J. Song, C. Qiu, X. Wang, X. Chen, Q. He, and H. Sheng, "Hastening stream offloading of inference via multi-exit dnns in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 1, pp. 535–548, 2024.

[18] Z. Huang, F. Dong, D. Shen, J. Zhang, H. Wang, G. Cai, and Q. He, "Enabling low latency edge intelligence based on multi-exit dnns in the wild," in *IEEE ICDCS*, 2021.

[19] W. Miao, Z. Zeng, L. Wei, S. Li, C. Jiang, and Z. Zhang, "Adaptive dnn partition in edge computing environments," in *IEEE ICPADS*, 2020.

[20] X. Wang, D. Wu, X. Wang, R. Zeng, L. Ma, and R. Yu, "Truthful auction-based resource allocation mechanisms with flexible task offloading in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 6377–6391, 2024.

[21] G. Li, J. Cai, X. Chen, and Z. Su, "Nonlinear online incentive mechanism design in edge computing systems with energy budget," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4086–4102, 2023.

[22] F. Wang, L. Jiao, K. Zhu, and L. Zhang, "Online edge computing demand response via deadline-aware v2g discharging auctions," *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 7279–7293, 2022.

[23] Q. Wang, S. Guo, J. Liu, C. Pan, and L. Yang, "Profit maximization incentive mechanism for resource providers in mobile edge computing," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 138–149, 2022.

[24] "State of IoT 2024," https://iot-analytics.com/number-connected-iot-devices/, 2024.

[25] "Home idle load," https://en.wikipedia.org/wiki/Home_idle_load, 2024.

[26] "Jetson Xavier NX Series," https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/, 2024.

[27] "HIKVISION," https://www.hikvision.com/cn/products/pc-products/, 2024.

[28] F. Cañellas, D. Camps-Mur, A. Fernández-Fernández, I. Boyano, M. Urias, J. Navarro-Ortiz, and J. J. Ramos-Muñoz, "5g nr, wi-fi and lifi multi-connectivity for industry 4.0," in *IEEE INFOCOM Workshops*, 2023.

[29] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023. [Online]. Available: https://www.mdpi.com/2504-4990/5/4/83

[30] F. Glover, "A multiphase-dual algorithm for the zero-one integer programming problem," *Operations Research*, vol. 13, no. 6, pp. 879–919, 1965.

[31] T. S. Breusch and A. R. Pagan, "The lagrange multiplier test and its applications to model specification in econometrics," *The review of economic studies*, vol. 47, no. 1, pp. 239–253, 1980.

[32] Y. Jin, L. Jiao, M. Ji, Z. Qian, S. Zhang, N. Chen, and S. Lu, "Scheduling in-band network telemetry with convergence-preserving federated learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 5, pp. 2313–2328, 2023.

[33] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[34] "Welcome to CVXPY 1.1," https://www.cvxpy.org, 2024.

[35] T. Achterberg, "What's new in gurobi 9.0," *Webinar Talk url: https://www. gurobi. com/wp-content/uploads/2019/12/Gurobi-90-Overview-Webinar-Slides-1. pdf*, vol. 5, no. 9, pp. 97–113, 2019.

[36] A. Archer and E. Tardos, "Truthful mechanisms for one-parameter agents," in *IEEE FOCS*, 2001.

[37] M. Ji, C. Su, Y. Fan, Y. Jin, Z. Qian, Y. Yan, Y. Chen, T. Cao, S. Zhang, and B. Ye, "Intaas: Provisioning in-band network telemetry as a service via online learning," *Computer Networks*, 2024.

[38] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6350–6364, 2017.

[39] "Amazon," https://aws.amazon.com/ec2/pricing/on-demand/, 2024.

[40] G. Singh, K. Bipin, and R. Dhawan, "Optimizing the boot time of android on embedded system," in *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*, 2011, pp. 503–508.

[41] K. Liu, A. Gurudutt, T. Kamaal, C. Divakara, and P. Prabhakaran, "Edge computing framework for distributed smart applications," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing and Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017, pp. 1–8.

[42] Y. Yan, S. Zhang, Y. Jin, F. Cheng, Z. Qian, and S. Lu, "Spatial and temporal detection with attention for real-time video analytics at edges," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2024.

[43] "Open-data-users," https://tfl.gov.uk/info-for/open-data-users/our-open-data, 2024.

[44] Y. Li, A. Zhou, X. Ma, and S. Wang, "Profit-aware edge server placement," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 55–67, 2022.

[45] "MTA Subway Hourly Ridership," https://data.ny.gov/Transportation/MTA-Subway-Hourly-Ridership-Beginning-July-2020/, 2024.

[46] S. Su, Z. Zhou, T. Ouyang, R. Zhou, and X. Chen, "Learning to be green: Carbon-aware online control for edge intelligence with colocated learning and inference," in *IEEE ICDCS*, 2023.

[47] "Planetlab," http://www.planet-lab.org/, 2024.

[48] "Seattle," https://seattle.poly.edu/, 2024.

[49] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[50] T. Ouyang, K. Zhao, X. Zhang, Z. Zhou, and X. Chen, "Dynamic edge-centric resource provisioning for online and offline services co-location," in *IEEE INFOCOM*, 2023.

[51] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.

[52] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[53] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[57] "Edge Device Memory," https://www.siemens.com/global/en/products/automation/topic-areas/industrial-edge/industrial-edge-devices.html#Contactourexperts, 2024.

[58] M. Ji, J. Qi, L. Jiao, G. Luo, H. Zhao, X. Li, B. Ye, and Z. Qian, "Cpn meets learning: Online scheduling for inference service in computing power network," *Computer Networks*, vol. 256, p. 110903, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128624007357

[59] D. Pariag and T. Brecht, "Application bandwidth and flow rates from 3 trillion flows across 45 carrier networks," in *PAM*, 2017.

[60] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. Lau, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2710–2721, 2013.

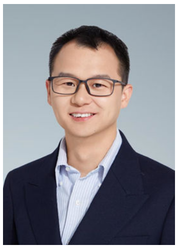
[61] D. Wen, P. Liu, G. Zhu, Y. Shi, J. Xu, Y. C. Eldar, and S. Cui, "Task-oriented sensing, computation, and communication integration for multi-device edge ai," *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 2486–2502, 2024.

Mingtao Ji received a B.E. degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics in 2018. He is currently pursuing the Ph.D. degree under the supervision of Professor Zhuzhong Qian in Nanjing University. To date, he has already published over 17 papers, including in journals such as IEEE TMC, IEEE/ACM TON, COMNET, JCST, and conferences such as IEEE INFOCOM, IEEE SECON, IEEE ICPADS, IEEE ISPA, IEEE ICC, IEEE/ACM IWQOS poster, and BigCom. He won the *Best Paper Award* in BigCom 2023. His research interests include Edge Inference, CPN, P4 switch, SDN, Federated Learning, Big Data Analytics, and Distributed Machine Learning.

Hehan Zhao received the HBSc degree from the Faculty of Arts & Science, University of Toronto in 2023. He is currently pursuing a Master of Engineering (MEng) in the Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto, with emphases in Computer Engineering and Data Analytics & Machine Learning. His research interests include edge computing, large language models (LLMs), natural language processing (NLP), and computer vision.

Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently a faculty member at the University of Oregon, USA and was previously a member of technical staff at Nokia Bell Labs, Ireland. He is interested in optimization, control, learning, and mechanism design applied to computer and telecommunication systems, networks, and services. He has published more than 80 papers in journals such as JSAC, TMC, ToN, and TPDS and conferences such as INFOCOM, MOBIHOC, ICDCS, SECON, and ICNP. He is a recipient of the U.S. National Science Foundation CAREER award, the Best Paper Awards of IEEE LANMAN 2013 and IEEE CNS 2019, and the 2016 Alcatel-Lucent Bell Labs UK and Ireland Recognition Award. He has been on the program committees of INFOCOM, MOBIHOC, and ICDCS.

Sheng Zhang is an associate professor at the School of Computer Science, Nanjing University. He is also a member of the State Key Lab. for Novel Software Technology. He received the BS and PhD degrees from Nanjing University in 2008 and 2014, respectively. His research interests include cloud computing and edge computing. To date, he has published more than 80 papers, including those appeared in TMC, TON, TPDS, TC, MobiHoc, ICDCS, INFOCOM, SECON, IWQoS and ICPP. He received the Best Paper Award of IEEE ICCCN 2020 and the Best Paper Runner-Up Award of IEEE MASS 2012. He is the recipient of the 2015 ACM China Doctoral Dissertation Nomination Award. He is a member of the IEEE and a senior member of the CCF.



Xin Li received the B.S. and Ph.D degrees from Nanjing University in 2008 and 2014, respectively. Currently, he is an associate professor in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include cloud computing, edge computing, and distributed computing.

Zhuzhong Qian is a professor at the School of Computer Science, Nanjing University, P. R. China. He received his PhD. Degree in computer science in 2007. Currently, his research interests include edge intelligence, datacenter networking and distributed machine learning. He has published over 80 research papers including JSAC, ToN, TPDS, INFOCOM, ICDCS, IPDPS and SECON. He received best paper candidates of WoWMoM 2021 and best paper runner-up of ICPADS 2021.

Baoliu Ye is a full professor at the School of Computer Science, Nanjing University. He received his Ph.D. in computer science from Nanjing University, China in 2004. He served as a visiting researcher of the University of Aizu, Japan from March 2005 to July 2006. His current research interests mainly include distributed systems, cloud computing, edge intelligence with over 100 papers published in major conferences and journals. He is the regent of CCF, vice director of CCF Technical Committee of Distributed Computing and Systems, and a member of IEEE.