Toward Market-Assisted AI: Cloud Inference for Streamed Data via Model Ensembles from Auctions

Yining Zhang, Lei Jiao, *Member, IEEE,* Konglin Zhu, Xiaojun Lin, *Fellow, IEEE,* Lin Zhang

***Abstract*—While ensemble methods can tackle concept drifts, obtaining pretrained models and conducting ensemble learning upon streamed data impose fundamental challenges, including the dynamic balance between system overhead and inference accuracy in uncertain system environments, and the interlacement between desired economic properties and long-term participation. In this paper, we propose the joint optimization which enables service providers to obtain models via repetitive auctions from the model providers and conduct ensemble methods online in a cost-efficient manner. We design polynomial-time online algorithms to solve the underlying non-linear mixed-integer social cost minimization problem, involving bid selection, payment allocation, model hosting, and ensemble model-weight adaption. We further rigorously prove the performance guarantees with our approach, such as the sub-linear dynamic regret for the bidding cost, the sub-linear dynamic fit for the long-term participation constraint, the truthfulness and the individual rationality for the auctions, the upper bound for ensemble inference loss, and the parameterized-constant competitive ratio for the long-term social cost. Through extensive trace-driven evaluations under real-world settings, we have validated the significant advantages of our approach over multiple baselines and state-of-the-art algorithms.**

***Index Terms*—Ensemble Learning, Inference, Data Stream, Cloud Computing, Online Optimization, Auction.**

I. Introduction

IT is very common for cloud services to perform machine learning inference in real time upon requests from end users [1], [2]. Often, as time goes, the inference accuracy of the *service provider*'s model may vary or even irreversibly decrease because of concept drifts [3], [4], i.e., the existing model can no longer infer the best label for the data. One solution is to retrain or dynamically update the model as concept drifts are detected. However, for real-time inference on users' request streams, this could be infeasible due to insufficient training data, long delay of training, or prohibitive computation overhead. In contrast, ensemble methods [5]–[7] can combine multiple complementary pretrained models, including the service provider's model and those provided by others (which we can call *model providers*), to achieve better accuracy and resilience against concept drifts without detecting them explicitly, thus providing an intriguing alternative.

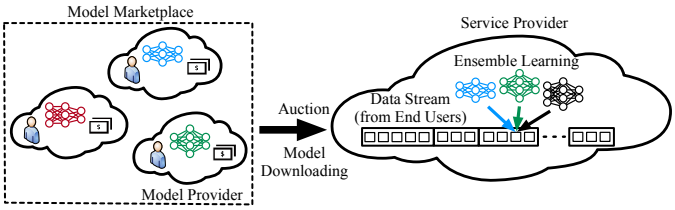

Fig. 1: System scenario

In fact, the service provider can engage in model transactions through AI model marketplaces with model providers [8]–[11]. To incentivize the model providers to contribute their models, a monetizing mechanism is needed. Some AI marketplaces adopt direct pricing [12], [13], which can be less ideal due to the risk of mispricing [14], the lack of market efficiency, and the inability to adapt to real-time demand and supply. *Auctions* can address these disadvantages. In this paper, we advocate an auction-based approach, where the service provider can be the auctioneer and purchase models from the model providers that act as the bidders.

Unfortunately, designing such an auction and control mechanism faces multiple challenges, as the service provider has to optimally operate the system both by assembling the ensemble and performing inference in the "front end", and by jointly conducting auctions and managing models in the "back end".

The first is how to control the dynamic balance, in an online manner, between system overhead and inference accuracy upon users' data stream as the system operates continuously. Combining more models to form the ensemble may lead to better inference accuracy, but having too many models could incur excessive model hosting cost. Whether to discard a model is also tricky as it involves "time-coupled" decisions [15]—keeping the model local could save the model downloading cost from model providers if this model is needed in the future, but it could cause waste if this model is not needed later. That is, the models need to be selected and deployed in terms of not only the incurred cost, but also the contribution to the inference loss of the ensemble. The unknown and unpredictable time-varying system environments and concept drifts, as well as the nonlinearity of both the cost and the inference loss functions, all escalate the difficulty and the complexity of the problem.

The other challenge is how to design the repetitive auctions that attain the desired economic properties, including truthfulness (i.e., a bidder maximizes its utility by not cheating about

This work was supported in part by the National Key Research and Development Program of China (2023YFB2704500), the Beijing Natural Science Foundation (4222033 and 9232008), the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, and the U.S. National Science Foundation (CNS-2047719, CNS-2225949, CNS-2113893, and CNS-2225950). (Corresponding authors: Lei Jiao, Konglin Zhu.)

Y. Zhang, K. Zhu, and L. Zhang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: {ynzhang, klzhu, zhanglin}@bupt.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. Lin is with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (e-mail: linx@ecn.purdue.edu). Part of this work was completed when he moved to The Chinese University of Hong Kong.

the bidding price) and individual rationality (i.e., a bidder has no loss on utility regardless of the auction outcome). The well-known Vickrey-Clarke-Groves (VCG) auctions [16] often require exactly solving the underlying social cost problem which can be intractable in our setting. The service provider may also desire to incentivize the long-term participation [17]–[20] of model providers across auctions. Without a sufficient number of participants, the service provider may struggle to gather good models; if a model provider rarely wins in the auctions, it may quit permanently. Consequently, selecting the winning bids needs to consider not only the current auction but also existing and future auctions in the long term—buying a bid now could be unnecessary if the future bids end up cheaper or offer better models; but not buying it now could force us to buy bids later, even if they turn out to be costly or inferior. This interlacement between long-term participation and economic properties has not been typically studied previously.

Existing research is not adequate for addressing the aforementioned challenges. The work in [21]–[26] studies ensemble methods, but generally neglects the system environment where such methods are executed. The work in [27]–[32] investigates ensemble learning specifically in cloud and/or edge environments, but neither targets an online and data streaming setting nor accounts for the auction mechanisms. The work on AI model markets [12], [13], [33]–[36] also falls short, either overlooking economic interactions or lacking consideration of ensembles or the joint optimization of system performance. See Section VII for our more detailed discussions.

In this paper, we model the joint cost and inference structure of an auction-assisted ensemble-learning service for streamed data samples. We formulate a social cost minimization problem over time for the service provider, considering model hosting cost, model downloading cost, ensemble inference loss, payment flows, and the utility of model providers. Our formulation features arbitrarily time-varying inputs, time-coupled switching-cost terms, and long-term constraints, capturing all the aforementioned challenges. Our problem is a non-linear mixed-integer program, and is unsurprisingly NP-hard.

We propose a universal mechanism to select winning bids, allocate payments, host models, and conduct ensemble learning, regardless of the types of the models that comprise the ensemble. To that end, we design a holistic set of polynomial-time online algorithms that work together. In each auction, first, given the model hosting decisions, we design the Online Fractional Algorithm (i.e., Algorithm 1) to make fractional bid selections based on alternate descent-ascent primal-dual steps for a reformulation that absorbs the long-term constraint into the objective by a well-designed proximal term [37], [38]. Next, we design the Randomized Rounding Algorithm (i.e., Algorithm 2) to convert such fractional decisions into more tangible integer bid selections in a randomized manner without violating the residual constraints. Further, we design the Payment Allocation Algorithm (i.e., Algorithm 3) to replace VCG and calculate payments to the bids based on the actual fractional decision and bidding price and the envisaged fractional decisions with alternative bidding prices. Then, we design the Model Hosting Algorithm (i.e., Algorithm 4) that ties the repetitive auctions over time and invokes Algorithms 1 and 2 to select and deploy models via comparing in real time the most recent downloading cost of switching from previous models to the current models versus the cumulative hosting cost of retaining the current models since the last switch operation. Finally, we design the Ensemble Learning Algorithm (i.e., Algorithm 5) to combine the inference results of the models from Algorithm 4 as a weighted sum to produce the joint result for each data sample, while updating the weights in an exponential manner as data samples arrive [39].

We provide a thorough analysis of our proposed algorithms by formally proving a series of theorems. We exhibit that the *dynamic regret* and the *dynamic fit* [38], [40] are sub-linear, i.e., as the total number of time slots grows, the difference between the time-averaged bidding cost incurred by our online approach and its time-averaged offline optimum gradually vanishes, so does the time-averaged violation of the long-term participation. We show that truthfulness and individual rationality are met in expectation. We further imply that the inference loss generated by our ensemble method is upper-bounded by a constant times that generated by the single best model out of our selected models. Based on that, we establish the *competitiveness*, i.e., the social cost over time incurred by our online approach is no greater than a constant times the sum of the offline optimal cost over time plus the cumulative inference loss of the single best model for each time slot.

Finally, we implement and test a classification service upon multiple real-world time-lapse data streams [41], [42] and benchmark datasets [43], [44], with up to 64 model providers at geographically-distributed clouds worldwide [45], [46] under realistic settings [17], [20], [47], [48]. Our evaluations reveal the following results: (i) Compared to the baselines of the Random method, the Greedy method, and the state-of-the-art-based DTEL^+ and DES^+ method, our approach reduces the long-term social cost on average by 70%, 37%, 35%, and 46%, respectively, with the empirical competitive ratio of $2.1 \sim 2.7$; (ii) Our approach achieves truthfulness and individual rationality in every auction; (iii) Our approach incurs the lowest dynamic regret and the lowest dynamic fit, which only grow slowly with time; (iv) Our approach attains an inference accuracy of about $2\% \sim 9\%$ higher than DTEL^+ and DES^+; (v) Most of our algorithms complete in a few seconds for each auction with the delay-insensitive payment calculated in less than a minute, meeting the needs in reality.

II. Model and Problem Formulation

A. System Modeling

We summarize all our major notations in Table I.

Machine Learning Service: We consider a machine learning service that continuously conducts inference upon a stream of data samples submitted by the end users. This machine learning service can be provisioned and operated by a *service provider* in the service provider's own data center or a public cloud. The data samples arrive dynamically over a series of consecutive time slots $\mathcal{T} = \{1, 2, \ldots, T\}$. For the time slot $t \in \mathcal{T}$, we use $\mathcal{M}^t = \{1, 2, \ldots, M^t\}$ to index the data samples that arrive during t. For the data sample $m \in \mathcal{M}^t$, we represent it as $\{a_m^t, b_m^t\}$, where a_m^t refers to its feature values and b_m^t refers to its ground-truth label.

TABLE I: NOTATIONS

Input	Description
$\mathcal{T}$	Set of time slots
$\mathcal{N}$	Set of model providers
$\mathcal{M}^t$	Set of data samples at time slot t
a_m^t	Feature of data sample m at t
b_m^t	Ground-truth label of data sample m at t
u_n^t	Cost of downloading model n at t
v_n^t	Cost of hosting model n at t
e^t	Cost of hosting the service provider's own model at t
Δ_n^t	Indicator of whether or not model n at t is the same as that at $t-1$
W	Minimal total number of models hosted per time slot
$h_n^t(\cdot)$	Decision function of model n at t
$h^t(\cdot)$	Decision function of the service provider's own model at t
c_n^t	Bidding price of bid n at t
Q^t	Service provider's budget for bid procurement at t
ϕ_n	Lower bound for cumulative participation of model provider n
Decision	**Description**
x_n^t	Whether or not model n wins the auction at t
y_n^t	Whether or not model n is hosted at t
z^t	Whether or not the service provider's own model is hosted at t
$\alpha_{n,m}^t$	Weight of model n for performing inference for data sample m at t
β_m^t	Weight of the service provider's own model for performing inference for the data sample m at t
p_n^t	Payment to bid n at t

Models and Costs: Besides its own model, the service provider can purchase models from the *model providers*, denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$, via the auction market as elaborated below and then download the purchased models for the local deployment. We consider each model provider offers one model; a model provider that offers multiple models can be treated as multiple "virtual" model providers. We use u_n^t to denote the cost (e.g., traffic, delay) of downloading the model from the model provider $n \in \mathcal{N}$, i.e., the model n, at the time slot t, and use v_n^t to denote the cost (e.g., resource, energy consumption) of hosting the model n at the service provider's facility at t. A model provider may offer different models as time goes, so we use $\Delta_n^t \in \{1, 0\}$ to imply whether or not the model n at t stays the same as at $t-1$. We also use e^t to denote the cost of hosting the service provider's own model at t. Without loss of generality, for the ensemble learning to be described next, we envisage that the service provider wants to host a minimum number $W \geq 1$ of models at any t, where W is set based on the needs and capacity of the service provider.

Model Auctions: The service provider acts as the auctioneer, and the model providers act as the bidders. At each time slot t, the auctioneer conducts an auction, as shown in Fig. 2:

- *Step 1:* Each bidder n submits a bid in the format of $\{c_n^t, h_n^t(\cdot), \Delta_n^t\}$. c_n^t is the bidding price, i.e., the amount of money the bid wants to charge for selling the offered model; $h_n^t(\cdot)$ is the decision function of the offered model; and Δ_n^t has already been explained in the above.
- *Step 2:* Collecting all the bids, the auctioneer decides the winning bids, the payment p_n^t to be made to each winning bid n (note that p_n^t does not necessarily equals c_n^t), and notifies the bidders of the auction outcome.
- *Step 3:* The auctioneer downloads the models from the winning bidder's facilities if the same model was not previously downloaded or if the model provider offers a different model compared to the last time the model was downloaded from this model provider.
- *Step 4:* The auctioneer sends the payment to each winning bidder.

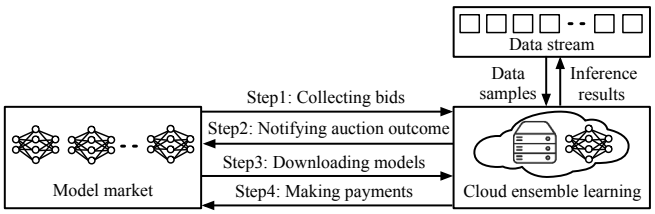

Fig. 2: System workflow in a single time slot

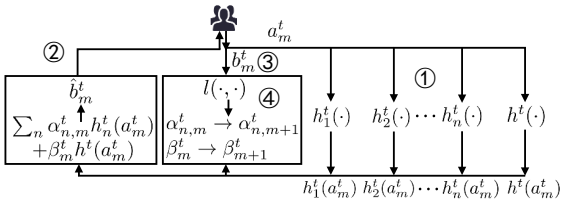

Fig. 3: Ensemble learning for a single data sample

To decide the winning bids, the auctioneer needs to solve the *social cost* minimization problem, which will be described next. For the auction at t, the auctioneer has the budget $Q^t \geq 0$, indicating the maximum number of models that can be purchased. To incentivize the long-term participation of model providers, we use $\phi_n \in [0, 1]$ to imply the participation threshold [17], [18]. That is, across all the T auctions, each model provider n should win for no less than $T \cdot \phi_n$ times; otherwise, we consider that the model provider n may quit the marketplace permanently. Step 4 may appear before Step 3, depending on the agreement in the market.

Ensemble Learning: At the time slot t, after deploying the models, the machine learning service employs *ensemble learning* to perform inference sequentially upon the data samples $\mathcal{M}^t$ by combining the inference results of different models. This workflow is as follows, shown in Fig. 3.

- *Step 1:* The data sample $m \in \mathcal{M}_t$ from an end user arrives with its feature a_m^t. The machine learning service uses each purchased model n and its own model, whose decision functions are $h_n^t(\cdot)$ and $h^t(\cdot)$, respectively, to make inference results $h_n^t(a_m^t)$ and $h^t(a_m^t)$.
- *Step 2:* The joint inference result is then produced as $\hat{b}_m^t = \sum_n \alpha_{n,m}^t h_n^t(a_m^t) + \beta_m^t h^t(a_m^t)$, where $\alpha_{n,m}^t$ is the weight for the purchased model n upon the data sample m and β_m^t is the weight for the service provider's own model upon the data sample m. $\hat{b}_m^t$ is further sent back to the corresponding end user.
- *Step 3:* The ground-truth label b_m^t arrives from the same end user.
- *Step 4:* The machine learning service updates the weights for all the models for the next data sample $m+1$.

The aforementioned process of conducting the inference first and receiving the ground truth afterward can capture many real-world applications such as ads recommendation systems [49], [50] and keyboard input methods [51]. Without loss of generality, we consider the squared loss function $l(\cdot, \cdot)$. Thus, we can denote the total cumulative loss incurred upon the entire data stream as $\sum_t \sum_m l(\hat{b}_m^t, b_m^t) = \sum_t \sum_m \left((\sum_n \alpha_{n,m}^t h_n^t(a_m^t) + \beta_m^t h^t(a_m^t)) - b_m^t\right)^2$.

Control Variables: The service provider needs to make the following control decisions. We use $x_n^t \in \{1, 0\}$ to denote

whether or not the model n wins the auction at the time slot t. We use $y_n^t \in \{1,0\}$ to denote whether or not the model n is hosted and deployed by the machine learning service at the time slot t. We use $z^t \in \{1,0\}$ to denote whether or not the machine learning service hosts and deploys its own model at the time slot t. We use $\alpha_{n,m}^t \in [0,1]$ to denote the weight of the purchased model n for performing inference for the data sample m at the time slot t. We use $\beta_m^t \in [0,1]$ to denote the weight of the service provider's own model for performing inference for the data sample m at the time slot t. We also use $p_n^t \geq 0$ to denote the payment made to the model provider n at the time slot t.

Cost of Auctioneer: The cost incurred at the machine learning service at t has the following components: (i) Cost of hosting models $\sum_n v_n^t y_n^t + e^t z^t$; (ii) Cost of downloading models $\sum_n (\Delta_n^t u_n^t [y_n^t - y_n^{t-1}]^+ + u_n^t y_n^t (1-\Delta_n^t))$, where $[\cdot]^+ = \max\{\cdot, 0\}$; (iii) Payment made to bidders $\sum_n x_n^t p_n^t$.

Cost of Bidders: The cost incurred at the model providers at t refers to the cost of training or producing the model (i.e., bidding price) determined by each model provider, minus any payment received from the service provider: $\sum_n x_n^t (c_n^t - p_n^t)$.

B. Problem Formulation and Algorithmic Challenges

Social Cost Minimization: The social cost refers to the sum of the cost of the service provider and the cost of the model providers over time, plus the cumulative inference loss upon the entire data stream. We formulate the social cost minimization problem $\mathbb{P}_0$ as follows. Note that the payment terms cancel one another, but the payments still need to be determined later to satisfy the desired *economic properties*.

$$\begin{aligned}
\mathbb{P}_0: \min & \textstyle\sum_t \sum_n x_n^t c_n^t + \sum_t \sum_n \Delta_n^t u_n^t \left[y_n^t - y_n^{t-1}\right]^+ \\
& + \textstyle\sum_t \sum_n y_n^t \left(v_n^t + u_n^t \left(1 - \Delta_n^t\right)\right) + \sum_t e^t z^t \\
& + \textstyle\sum_t \sum_m \left(\left(\sum_n \alpha_{n,m}^t h_n^t(a_m^t) + \beta_m^t h^t(a_m^t)\right) - b_m^t\right)^2, \qquad (1) \\
s.t. \quad & \textstyle\sum_n y_n^t + z^t \geq W, \forall t \in \mathcal{T}, \qquad (1a) \\
& \textstyle\sum_n x_n^t \leq Q^t, \forall t \in \mathcal{T}, \qquad (1b) \\
& \textstyle\sum_n \alpha_{n,m}^t + \beta_m^t = 1, \forall m \in \mathcal{M}^t, \forall t \in \mathcal{T}, \qquad (1c) \\
& \alpha_{n,m}^t \leq y_n^t, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}^t, \forall t \in \mathcal{T}, \qquad (1d) \\
& \beta_m^t \leq z^t, \forall m \in \mathcal{M}^t, \forall t \in \mathcal{T}, \qquad (1e) \\
& x_n^t \geq y_n^t, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}, \qquad (1f) \\
& \tfrac{1}{T} \textstyle\sum_t x_n^t \geq \phi_n, \forall n \in \mathcal{N}, \qquad (1g) \\
var. \quad & x_n^t, y_n^t, z^t \in \{0,1\}, \alpha_{n,m}^t, \beta_m^t \in [0,1], \\
& \forall n \in \mathcal{N}, \forall m \in \mathcal{M}^t, \forall t \in \mathcal{T}.
\end{aligned}$$

The objective (1) minimizes the social cost. Constraint (1a) ensures the minimum number of models for ensemble learning. Constraint (1b) enforces the purchase budget. Constraint (1c) normalizes the weights, without loss of generality. Constraint (1d) and (1e) ensure that the weight can be non-zero only if the corresponding model is hosted. Constraint (1f) ensures that a model can be hosted only if the corresponding bid wins in the auction. Constraint (1g) ensures the long-term participation. The domains for the decision variables are specified finally.

Our Mechanism

for $t = 1$ **to** T **do**

▷ **Model Hosting and Winning-Bid Selection:**

Invoke **Algorithm 4** to purchase and host models, which invokes **Algorithm 1** and **Algorithm 2**;

▷ **Ensemble Learning:**

Invoke **Algorithm 5** to obtain weights for the models and conduct inference upon the data stream;

▷ **Payment Allocation:**

Invoke **Algorithm 3** to calculate payments to the bids;

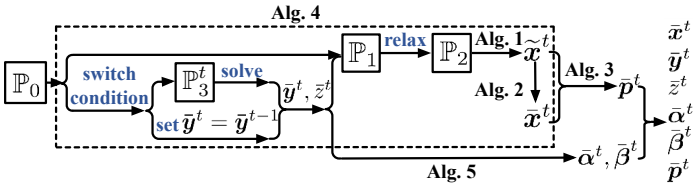

Fig. 4: Structure of our proposed approach

Algorithmic Challenges: Solving the problem $\mathbb{P}_0$ online is non-trivial due to multiple fundamental challenges. First, the *switching cost* term $\sum_t \sum_n \Delta_n^t u_n^t [y_n^t - y_n^{t-1}]^+$ couples every time slot $t-1$ and its next time slot t. At $t-1$, as we have no idea whether we will want to host the model at t, it is difficult to decide $\boldsymbol{y}^{t-1}$ to optimize the switching cost between $t-1$ and t. Second, the *long-term constraint* of participation $\sum_t x_n^t \geq \sum_t \phi_n$ needs to be considered carefully. Determining $\boldsymbol{x}^t$ at the current time slot t affects not only the cost at t but also the determination of $\boldsymbol{x}^t$ for future t, in order to satisfy this constraint. Third, our problem is nonlinear and NP-hard. If we ignore all the terms related to $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and the switching cost, our problem could then become the minimum-cost knapsack problem, which is known to be NP-hard. Solving an NP-hard problem on the fly as the inputs gradually reveal themselves can be more challenging. Fourth, we need to decide each winner's payment to satisfy the economic properties of truthfulness and individual rationality, as it is intertwined with the previous three challenges.

Algorithms Roadmap: Our mechanism design incorporates a set of polynomial-time online algorithms to overcome the aforementioned challenges, as shown in Fig. 4.

Section III describes our Algorithms 1, 2, and 3. Algorithm 1 outputs the fractional bid-selection decisions while accommodating the long-term constraint, and Algorithm 2 converts such fractional decisions into integers. Algorithm 3 further determines the payment for each bid. In this section, we define and prove the dynamic regret and the dynamic fitness with Algorithms 1 and 2, and also define and prove the truthfulness and the individual rationality with Algorithm 3.

Section IV describes our Algorithms 4 and 5. Algorithm 4 makes dynamic trade-offs between switching to a new decision and keeping the previous decision regarding model hosting, while actually invoking Algorithms 1 and 2 for bid selections. Algorithm 5 updates the weights for all the deployed models while conducting ensemble learning upon the data stream. In this section, we prove the competitive ratio of our entire approach, and also prove the loss bound of ensemble learning.

III. Bid Selection and Payment Allocation

A. Algorithm 1: Selecting Fractional Bids

We note that if $\boldsymbol{y}^t$ is given (which will be further described in Section IV-A), then we can actually extract the following problem $\mathbb{P}_1$ from the original problem $\mathbb{P}_0$:

$$\begin{aligned}
\mathbb{P}_1: \min & \textstyle\sum_t \sum_n x_n^t c_n^t, && (2)\\
s.t. \quad & x_n^t \geq y_n^t, \forall n \in \mathcal{N}, t \in \mathcal{T}, && (2a)\\
& \textstyle\sum_n x_n^t \leq Q^t, \forall t \in \mathcal{T}, && (2b)\\
& \tfrac{1}{T}\textstyle\sum_t x_n^t \geq \phi_n, \forall n \in \mathcal{N}, && (2c)\\
\text{var.} \quad & x_n^t \in \{0,1\}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}.
\end{aligned}$$

For a concise presentation, we denote the objective function of $\mathbb{P}_1$ as $f^t(\boldsymbol{x}^t) = \sum_n x_n^t c_n^t$. We also denote $g_n^t = \phi_n - x_n^t, \forall n \in \mathcal{N}$; $\mathbf{g}^t(\boldsymbol{x}^t) = [g_1^t, \ldots, g_N^t]$; $h_n^t = x_n^t - y_n^t, \forall n \in \mathcal{N}$; $\mathbf{h}^t(\boldsymbol{x}^t) = [h_1^t, \ldots, h_N^t]$; and $d^t(\boldsymbol{x}^t) = \sum_n x_n^t - Q^t$. We can further relax the bid-selection control variables to the real domain, and thus transform $\mathbb{P}_1$ to a linear program. That is,

$$\begin{aligned}
\mathbb{P}_2: \min & \textstyle\sum_t f^t(\boldsymbol{x}^t), && (3)\\
s.t. \quad & \mathbf{h}^t(\boldsymbol{x}^t) \succeq \mathbf{0}, && (3a)\\
& d^t(\boldsymbol{x}^t) \leq 0, && (3b)\\
& \textstyle\sum_t \mathbf{g}^t(\boldsymbol{x}^t) \preceq \mathbf{0}, && (3c)\\
\text{var.} \quad & \boldsymbol{x}^t \in \mathcal{X} = \{\boldsymbol{x}^t | x_n^t \in [0,1], \forall n \in \mathcal{N}, \forall t \in \mathcal{T}\}.
\end{aligned}$$

Solving the above problem $\mathbb{P}_2$ is equivalent to solving its min-max version as follows:

$$\begin{aligned}
\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} & \textstyle\sum_t \mathcal{L}^t(\boldsymbol{x}^t, \boldsymbol{\lambda}^t) = \sum_t f^t(\boldsymbol{x}^t) + \sum_t \boldsymbol{\lambda}^t \mathbf{g}^t(\boldsymbol{x}^t), && (4)\\
s.t. \quad & \mathbf{h}^t(\boldsymbol{x}^t) \succeq \mathbf{0}, d^t(\boldsymbol{x}^t) \leq 0, \boldsymbol{x}^t \in \mathcal{X},
\end{aligned}$$

where $\boldsymbol{\lambda}^t$ is the Lagrange multiplier at t, and the long-term constraint (3c) has been absorbed into the objective so that we do not have to worry about it at this point.

Based on (4), we can now solve $\mathbb{P}_2$ on the fly at each $t+1$ by a standard dual ascent step to update the dual variable $\boldsymbol{\lambda}^{t+1}$ as in (5), and a modified descent step to minimize $\mathcal{L}^t(\boldsymbol{x}, \boldsymbol{\lambda}^{t+1})$ and obtain the fractional solution $\widetilde{\boldsymbol{x}}^{t+1}$ as in (6).

$$\boldsymbol{\lambda}^{t+1} = [\boldsymbol{\lambda}^t + \eta \nabla_{\boldsymbol{\lambda}} \mathcal{L}^t(\widetilde{\boldsymbol{x}}^t, \boldsymbol{\lambda}^t)]^+, \tag{5}$$

where $\eta > 0$ is the step size; $\nabla_{\boldsymbol{\lambda}} \mathcal{L}^t(\widetilde{\boldsymbol{x}}^t, \boldsymbol{\lambda}^t)$ is the gradient of $\mathcal{L}^t(\widetilde{\boldsymbol{x}}^t, \boldsymbol{\lambda})$ given $\boldsymbol{\lambda} = \boldsymbol{\lambda}^t$; and $\widetilde{\boldsymbol{x}}^t$ is the fractional solution previously solved at t.

$$\begin{aligned}
\min & \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\boldsymbol{x} - \widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1} \mathbf{g}^t(\boldsymbol{x}) + \tfrac{\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma}, && (6)\\
s.t. \quad & \mathbf{h}^t(\boldsymbol{x}) \succeq \mathbf{0}, d^t(\boldsymbol{x}) \leq 0, \boldsymbol{x} \in \mathcal{X},
\end{aligned}$$

where γ is a predefined constant and $\nabla f^t(\widetilde{\boldsymbol{x}}^t)$ is the gradient of $f^t(\boldsymbol{x})$ at $\boldsymbol{x} = \widetilde{\boldsymbol{x}}^t$. This can be called a Modified Online Saddle-Point (MOSP) method [38]. MOSP differs from Online Mirror Descent (OMD), since standard OMD does not involve primal-dual updates [52], [53], but primal-dual-related OMD and its variants do not often capture either the constraint violation or the long-term time-varying constraints [54]–[60]. MOSP also relates to constrained online learning, as the latter focuses on the long-term time-invariant constraints [61]–[64] or the static regret [65]–[68].

Algorithm 1: Online Fractional Algorithm

Input: Fractional solution $\widetilde{\boldsymbol{x}}^t$, dual solution $\boldsymbol{\lambda}^t$, and $\boldsymbol{y}^t$
Output: Fractional solution $\widetilde{\boldsymbol{x}}^{t+1}$

1 **Initial:** $\boldsymbol{\lambda}^1 = \mathbf{0}$, $\eta > 0$
2 Calculate $\boldsymbol{\lambda}^{t+1}$ based on (5);
3 Calculate $\widetilde{\boldsymbol{x}}^{t+1}$ by solving the problem (6) optimally;

Algorithm 2: Randomized Rounding Algorithm

Input: Fractions $[\widetilde{x}_1^t, \ldots, \widetilde{x}_N^t]$
Output: Integers $[\bar{x}_1^t, \ldots, \bar{x}_N^t]$

1 ▷ Fractions scaling
2 Denote $\widetilde{\mathbf{W}}^t = [\widetilde{x}_1^t, \ldots, \widetilde{x}_N^t]$, $\varepsilon = \mathbf{1}^\top \widetilde{\mathbf{W}}^t$;
3 $\omega_1 = \frac{\lceil \varepsilon \rceil}{\varepsilon}$, $\omega_2 = \frac{\lfloor \varepsilon \rfloor}{\varepsilon}$;
4 $\widetilde{\mathbf{X}'}^t =$
$$\begin{cases} [\omega_1 \cdot \widetilde{x}_1^t, \ldots, \omega_1 \cdot \widetilde{x}_N^t] \ with \ probability \ \varepsilon - \lfloor \varepsilon \rfloor \\ [\omega_2 \cdot \widetilde{x}_1^t, \ldots, \omega_2 \cdot \widetilde{x}_N^t] \ with \ probability \ \lceil \varepsilon \rceil - \varepsilon \end{cases}$$
5 Define $\mathcal{N}' = \mathcal{N} \backslash \{n | \widetilde{x'}_n^t \in \{0,1\}\}$;
6 ▷ Fractions rounding
7 **while** $\mathcal{N}' \neq \emptyset$ **do**
8 Select $n_1, n_2 \in \mathcal{N}'$, $n_1 \neq n_2$;
9 $\varphi_1 = \min\{1 - \widetilde{x'}_{n_1}^t, \widetilde{x'}_{n_2}^t\}, \varphi_2 = \min\{1 - \widetilde{x'}_{n_2}^t, \widetilde{x'}_{n_1}^t\}$;
10 $(\widetilde{x''}_{n_1}^t, \widetilde{x''}_{n_2}^t) =$
$$\begin{cases} (\widetilde{x'}_{n_1}^t + \varphi_1, \widetilde{x'}_{n_2}^t - \varphi_1) \ with \ probability \ \frac{\varphi_2}{\varphi_1 + \varphi_2} \\ (\widetilde{x'}_{n_1}^t - \varphi_2, \widetilde{x'}_{n_2}^t + \varphi_2) \ with \ probability \ \frac{\varphi_1}{\varphi_1 + \varphi_2} \end{cases}$$
11 **if** $\widetilde{x''}_{n_1}^t \in \{0,1\}$ **then** $\bar{x}_{n_1}^t = \widetilde{x''}_{n_1}^t, \mathcal{N}' = \mathcal{N}' \backslash \{n_1\}$;
12 **else** set $\widetilde{x'}_{n_1}^t = \widetilde{x''}_{n_1}^t$;
13 **if** $\widetilde{x''}_{n_2}^t \in \{0,1\}$ **then** $\bar{x}_{n_2}^t = \widetilde{x''}_{n_2}^t, \mathcal{N}' = \mathcal{N}' \backslash \{n_2\}$;
14 **else** set $\widetilde{x'}_{n_2}^t = \widetilde{x''}_{n_2}^t$;

Algorithm 1 follows exactly the above idea. Through such alternate descent-ascent steps, we solve for $\boldsymbol{x}^{t+1}$ and $\boldsymbol{\lambda}^{t+1}$ at each $t+1$ using only the inputs that have been known before $t+1$ rather than the unknowable future information beyond (and including) $t+1$. $\mathcal{L}^t(\boldsymbol{x}, \boldsymbol{\lambda}^{t+1})$ is approximated by $\nabla f^t(\widetilde{\boldsymbol{x}}^t)(\boldsymbol{x} - \widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1} \mathbf{g}^t(\boldsymbol{x})$, and $\frac{1}{2\gamma}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}^t\|^2$ is a regularization and proximal term. The problem (6) is solvable using standard convex optimization solvers which can find the ν-accurate optimal solution in $O(N^2 \log(1/\nu))$ time [69] via the interior-point method, for example.

B. Algorithm 2: Converting Fractional Bids to Integral Bids

Algorithm 2 converts the fractional bid-selection decisions from Algorithm 1 into integers through a randomized rounding process without violating the constraints (2a) and (2b) during rounding while ensuring the expectation of each integer after rounding equals the corresponding fraction before rounding. Such expectation preservation is important for Algorithm 3.

Fractions scaling as in Lines 2∼5 adjusts the fractions and ensures that the sum of the fractions after adjusting equals an integer. In Line 4, after adjusting, each column of $\widetilde{\mathbf{W}}^t$ either increases by multiplying $\omega_1 > 1$ or decreases by multiplying

$\omega_2 < 1$. Thus, we have $E[\widetilde{\mathbf{X}'}^t] = \frac{\lceil\varepsilon\rceil}{\varepsilon}\widetilde{\mathbf{W}}^t \cdot (\varepsilon - \lfloor\varepsilon\rfloor) + \frac{\lfloor\varepsilon\rfloor}{\varepsilon}\widetilde{\mathbf{W}}^t \cdot (\lceil\varepsilon\rceil - \varepsilon) = (\lceil\varepsilon\rceil - \lfloor\varepsilon\rfloor)\widetilde{\mathbf{W}}^t = \widetilde{\mathbf{W}}^t$. We should mention that the sum of $\widetilde{x}_n^t$ increases to at most $\lceil\varepsilon\rceil$, and Q^t is an integer, so (2b) still holds after adjusting; and (2a) always holds.

Fractions rounding as in Lines 7∼14 iteratively selects a pair of fractions and converts at least one of them into an integer in each iteration. Since the sum of all columns is an integer beforehand due to the previous step, it can be guaranteed that $\widetilde{\mathbf{X}'}^t$ as a vector only contains 0 and 1 after the loop. In Line 10, the sum is preserved, because $\widetilde{x''}^t_{n_1} + \widetilde{x''}^t_{n_2} = (\widetilde{x'}^t_{n_1} + \varphi_1) + (\widetilde{x'}^t_{n_2} - \varphi_1) = \widetilde{x'}^t_{n_1} + \widetilde{x'}^t_{n_2}$, for example. Also, the expectation is preserved. That is, for every n, for the first case in Line 10, we have $E(\widetilde{x''}^t_n) = (\varepsilon - \lfloor\varepsilon\rfloor)\frac{\varphi_2}{\varphi_1+\varphi_2}(\omega_1\widetilde{x}_n^t + \varphi_1) + (\varepsilon - \lfloor\varepsilon\rfloor)\frac{\varphi_1}{\varphi_1+\varphi_2}(\omega_1\widetilde{x}_n^t - \varphi_2) + (\lceil\varepsilon\rceil - \varepsilon)\frac{\varphi_2}{\varphi_1+\varphi_2}(\omega_2\widetilde{x}_n^t + \varphi_1) + (\lceil\varepsilon\rceil - \varepsilon)\frac{\varphi_1}{\varphi_1+\varphi_2}(\omega_2\widetilde{x}_n^t - \varphi_2) = \widetilde{x}_n^t$; and it is analogous for the second case. We note that Lines 9 and 10 guarantee that in each iteration of the loop at least one fraction will be rounded to either 0 or 1. This fractions rounding phase borrows the idea of dependent rounding [70]. The time complexity of Algorithm 2 is $O(N)$.

C. Algorithm 3: Allocating Payment

Algorithm 3 calculates the payments to the winning bids output by Algorithm 2. Following the sufficient and necessary conditions for a randomized auction to be both truthful and individually rational [71], which will be formally defined in Section III-E, we use Line 2 to calculate the payment to the winning bids. Our auction at each time slot t has become a randomized auction due to Algorithm 2. We note that the fractional solution $\widetilde{x}_n^t$ can be considered as a function of c_n^t and $\boldsymbol{c}_{-n}^t$, where the former is the bidding price reported by the bid n and the latter is what is reported by all the other bids, because the value of $\widetilde{x}_n^t$ is calculated by Algorithm 1 using c_n^t and $\boldsymbol{c}_{-n}^t$ as the inputs. In other words, Algorithm 1 can be considered as the mapping from c_n^t and $\boldsymbol{c}_{-n}^t$ to $\widetilde{x}_n^t$. In a function format, we can just write $\widetilde{x}_n^t(c_n^t, \boldsymbol{c}_{-n}^t)$. In Line 2 of Algorithm 3, to calculate the integral, we need to vary the value of c_n^t, simply denoted as c here, keep $\boldsymbol{c}_{-n}^t$, and invoke Algorithm 1 to calculate the corresponding $\widetilde{x}_n^t$. The integration interval's upper limit χ_n^t needs to be set to satisfy the requirements in Theorem 2 which will be stated later. Specifically, we need to find out the value of χ_n^t for n at t, such that $\int_0^{\chi_n^t} \widetilde{x}_n^t(c, \boldsymbol{c}_{-n}^t)dc < \infty$ and $\int_{\chi_n^t}^{\infty} \widetilde{x}_n^t(c, \boldsymbol{c}_{-n}^t)dc = 0$. Toward that end, we focus on the problem (6), whose objective function is as follows, where x_n^{t+1}, $\forall n$ are the decision variables and λ_n^{t+1}, $\forall n$ come from (5):

$$\sum_n \left(c_n^t(x_n^{t+1} - \widetilde{x}_n^t) + \lambda_n^{t+1}(\phi_n - 2x_n^{t+1} + y_n^{t+1}) + \frac{(x_n^{t+1} - \widetilde{x}_n^t)^2}{2\gamma} \right).$$

Note that this is a summation of a series of quadratic functions. For each n, we desire that, in the interval $[0, 1]$, its optimal x_n^{t+1} is 0. From the geometric image, we know that this can only be achieved when the vertical symmetry axis of the quadratic function falls on the non-positive half of the horizontal real axis. That is, we require $\widetilde{x}_n^t - \gamma c_n^t + 2\gamma\lambda_n^{t+1} \leq 0$, i.e., $c_n^t \geq \frac{\widetilde{x}_n^t}{\gamma} + 2\lambda_n^{t+1}$. Then we conclude that $\chi_n^t = \frac{\widetilde{x}_n^t}{\gamma} + 2\lambda_n^{t+1}$.

Algorithm 3: Payment Allocation Algorithm

Input: $[\bar{x}_1^t, \ldots, \bar{x}_N^t]$
Output: Payment $p_n^t, \forall n$
1 **for** *n, where* $\bar{x}_n^t = 1$ **do**
2 Set $p_n^t = c_n^t \widetilde{x}_n^t(c_n^t, \boldsymbol{c}_{-n}^t) + \int_{c_n^t}^{\chi_n^t} \widetilde{x}_n^t(c, \boldsymbol{c}_{-n}^t)dc$, where $\chi_n^t = \frac{\widetilde{x}_n^t}{\gamma} + 2\lambda_n^{t+1}$;
3 **for** *n, where* $\bar{x}_n^t = 0$ **do**
4 Set $p_n^t = 0$;

The time complexity of Algorithm 3 is $O(N^3 i \log(1/\nu))$, where i is the number of segments when dividing the range of $\chi_n^t - c_n^t$ for calculating the integral numerically.

D. Regret and Fit Analysis

We define and adopt the *dynamic regret* and the *dynamic fit* as the performance metrics. We prove that the dynamic regret and the dynamic fit of our algorithms grow only sub-linearly along with the length of the entire time horizon.

Definition 1 (Dynamic Regret and Dynamic Fit) *The dynamic regret and the dynamic fit for the problem $\mathbb{P}_1$ are defined as follows, respectively:*

$$\mathrm{Reg}^T := E[\textstyle\sum_{t=1}^T f^t(\bar{\boldsymbol{x}}^t)] - \sum_{t=1}^T f^t(\boldsymbol{x}^{t*}),$$
$$\mathrm{Fit}^T := \|[E[\textstyle\sum_{t=1}^T \mathbf{g}^t(\bar{\boldsymbol{x}}^t)]]^+\|,$$

where for each t, $\bar{\boldsymbol{x}}^t$ is the output of Algorithm 2 as described previously; $\boldsymbol{x}^{t*} \in \arg\min_{\boldsymbol{x}^t \in \mathcal{X}^t} f^t(\boldsymbol{x}^t)$; $\mathcal{X}^t := \{\boldsymbol{x} | \mathbf{h}^t(\boldsymbol{x}) \succeq \mathbf{0}, d^t(\boldsymbol{x}) \leq 0, \mathbf{g}^t(\boldsymbol{x}) \preceq \mathbf{0}; x_n^t \in \{0, 1\}, \forall n\}$.

While the dynamic regret captures the difference between the objective value incurred by our online solution and the sum of the optimal objective value at each time slot, the dynamic fit reflects the violation of Constraint (2c). Recall that (2c) was absorbed into the objective in Algorithm 1; so it is important to quantify the violation of this constraint. We actually have the following result:

Theorem 1 *The dynamic regret and the dynamic fit satisfy $\mathrm{Reg}^T \leq \mathcal{O}(T^{\frac{2}{3}})$ and $\mathrm{Fit}^T \leq \mathcal{O}(T^{\frac{2}{3}})$.*

Proof. See Appendix A in our supplementary material. □

Theorem 1 relies on some common assumptions, including the requirement that the variation in the environment is sub-linear, i.e., the cumulative variations of the per-time-slot minimizers and of the per-time-slot constraints grow sub-linearly with time, respectively. Note that the supplementary material is a separate document along with this paper.

E. Economic Properties Analysis

We define *utility*, based on which we further define *truthfulness* and *individual rationality*. We prove that our randomized auction mechanism satisfies both of these economic properties.

Definition 2 (Utility) *The utility of the bid n at time t is*

$$U_n(c_n^t, \boldsymbol{c}_{-n}^t) = \begin{cases} p_n^t(\hat{c}_n^t, \hat{\boldsymbol{c}}_{-n}^t) - c_n^t E(\bar{x}_n^t(\hat{c}_n^t, \hat{\boldsymbol{c}}_{-n}^t)), \\ \quad \textit{if } \bar{x}_n^t = 1 \\ 0 \textit{ , otherwise} \end{cases}$$

Algorithm 4: Model Hosting Algorithm

Output: $\bar{\boldsymbol{y}}^t$, $\bar{z}^t$
1 **if** $t = 1$ **then**
2 Initialize $\hat{t} = 1$, $\bar{\boldsymbol{y}}^0$;
3 Obtain the optimal solution $\widehat{\boldsymbol{y}}^1$, $\bar{z}^1$ to $\mathbb{P}_3^1$;
4 Given $\bar{\boldsymbol{y}}^1 = \widehat{\boldsymbol{y}}^1$, obtain $\bar{\boldsymbol{x}}^1$ by **Algorithms 1** and **2**;
5 **else**
6 **if** $k \cdot C_{SC}^{\hat{t}}(\bar{\boldsymbol{y}}^{\hat{t}}, \bar{\boldsymbol{y}}^{\hat{t}-1}) \leq \sum_{\tau=\hat{t}}^{t-1} C_{-SC}^{\tau}(\bar{\boldsymbol{x}}^{\tau}, \bar{\boldsymbol{y}}^{\tau}, \bar{z}^{\tau})$ **then**
7 Obtain the optimal solution $\widehat{\boldsymbol{y}}^t$, $\bar{z}^t$ to $\mathbb{P}_3^t$;
8 Given $\bar{\boldsymbol{y}}^t = \widehat{\boldsymbol{y}}^t$, get $\bar{\boldsymbol{x}}^t$ by **Algorithms 1** and **2**;
9 **if** $\bar{\boldsymbol{y}}^t \neq \bar{\boldsymbol{y}}^{t-1}$ **then**
10 $\hat{t} = t$;
11 **if** $\hat{t} < t$ **then**
12 $\bar{\boldsymbol{y}}^t = \bar{\boldsymbol{y}}^{t-1}$;
13 Given $\bar{\boldsymbol{y}}^t$, obtain $\bar{\boldsymbol{x}}^t$ by **Algorithms 1** and **2** and obtain $\bar{z}^t$ by solving $\mathbb{P}_3^t$;

where $\hat{c}_n^t$ is the bidding price submitted by the bid n; $\hat{\boldsymbol{c}}_{-n}^t$ denotes the bidding prices of all the other bids except the bid n; and c_n^t is the true cost of the bid n.

Definition 3 (Truthfulness) *A randomized auction is truthful in expectation if every bid n maximizes its expected utility by bidding its true cost, i.e., $U_n^t(c_n^t, \hat{\boldsymbol{c}}_{-n}^t) \geq U_n^t(\hat{c}_n^t, \hat{\boldsymbol{c}}_{-n}^t)$.*

Definition 4 (Individual Rationality) *A randomized auction is individually rational in expectation if every bid n always achieves a non-negative utility, i.e., $U_n^t(\hat{c}_n^t, \hat{\boldsymbol{c}}_{-n}^t) \geq 0$.*

Theorem 2 *A randomized auction is truthful and individually rational in expectation if and only if the following conditions are met [71]: (i) $E(\bar{x}_n^t)$ is monotonically non-increasing in $c_n^t, \forall n$; (ii) $\int_0^{\infty} E(\bar{x}_n^t)dc < \infty, \forall n$; (iii) the payment is in the form of $p_n^t = c_n^t E(\bar{x}_n^t(c_n^t, \boldsymbol{c}_{-n}^t)) + \int_{c_n^t}^{\infty} E(\bar{x}_n^t(c, \boldsymbol{c}_{-n}^t))dc, \forall n$. Our randomized auction mechanism satisfies these conditions.*

Proof. See Appendix B in our supplementary material. □

The two economic properties are important because truthfulness encourages the bidder to report its true cost as the bidding price, and individual rationality guarantees no utility loss for the bidder regardless of the auction outcome. We note that Definitions 2∼4 and Theorem 2 only deal with $E(\bar{x}_n^t)$; yet, via Algorithms 1 and 2, we are able to establish $E(\bar{x}_n^t) = \widetilde{x}_n^t$ and further prove that the conditions as required in Theorem 2 can be met by using $E(\bar{x}_n^t) = \widetilde{x}_n^t$. Following Theorem 2, we can design Algorithm 3.

IV. Model Hosting and Ensemble Learning

A. Algorithm 4: Hosting Models via Lazy Switch

Algorithm 4 neglects the inference loss tentatively and dynamically determines the changing set of the models to host at the service provider's facility as time goes. Our idea is to keep hosting the current set of models until the cumulative model hosting cost exceeds a pre-specified constant times the most recent switching cost of switching from a previous set of models to the current set of models. To determine the set of models that we would want to potentially switch to, we use $\mathbb{P}_3^t$ at each t, which is extracted from our original problem $\mathbb{P}_0$.

Algorithm 5: Ensemble Learning Algorithm

Input: $\bar{\boldsymbol{y}}^t$, $\bar{z}^t$, data samples $\mathcal{M}^t$, step size μ_m^t, $\forall m$
Output: Inferred labels $\hat{b}_m^t, \forall m$
1 Initialize weight parameters: $\xi_{n,1}^t = \bar{y}_n^t$, $\psi_1^t = \bar{z}^t$
2 **for** $m = 1$ *to* M^t **do**
3 ▷ Weights update
4 **for** $n = 1$ *to* N **do**
5 $\alpha_{n,m}^t = \begin{cases} \xi_{n,m}^t / (\sum_n \xi_{n,m}^t + \psi_m^t), & \bar{y}_n^t = 1 \\ 0, & \bar{y}_n^t = 0 \end{cases}$
6 $\beta_m^t = \begin{cases} \psi_m^t / (\sum_n \xi_{n,m}^t + \psi_m^t), & \bar{z}^t = 1 \\ 0, & \bar{z}^t = 0 \end{cases}$
7 ▷ Label inference
8 $\hat{b}_m^t = \sum_n \alpha_{n,m}^t h_n^t(a_m^t) + \beta_m^t h^t(a_m^t)$;
9 ▷ Weight parameters update
10 receive the ground-truth label b_m^t;
11 **for** $n = 1$ *to* N **do**
12 $\xi_{n,m+1}^t = \xi_{n,m}^t \exp\{-\mu_m^t l(h_n^t(a_m^t), b_m^t)\}$;
13 $\psi_{m+1}^t = \psi_m^t \exp\{-\mu_m^t l(h^t(a_m^t), b_m^t)\}$;

$$\mathbb{P}_3^t: \min \sum_n y_n^t (v_n^t + u_n^t(1 - \Delta_n^t)) + e^t z^t, \tag{7}$$
$$s.t. \quad \sum_n y_n^t \leq Q^t, \tag{7a}$$
$$\sum_n y_n^t + z^t \geq W, \tag{7b}$$
$$\text{var.} \quad y_n^t, z^t \in \{0, 1\}, \forall n \in \mathcal{N}.$$

The coefficient matrix of the constraints of $\mathbb{P}_3^t$ is totally unimodular [72]. Therefore, we can solve $\mathbb{P}_3^t$ with its decision variables in $[0, 1]$ via any standard linear program solver, and the optimal solution we obtain will be automatically in $\{0, 1\}$.

Algorithm 4 adopts the above idea in Line 6, where k is the pre-specified parameter. We also define $C_{SC}^t(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) = \sum_n \Delta_n^t u_n^t [y_n^t - y_n^{t-1}]^+$ and $C_{-SC}^t(\boldsymbol{x}^t, \boldsymbol{y}^t, z^t) = \sum_n x_n^t c_n^t + \sum_n y_n^t (v_n^t + u_n^t(1 - \Delta_n^t)) + e^t z^t$. $\hat{t}$ records the most recent time slot where a switch operation occurs. In Line 7, we solve $\mathbb{P}_3^t$ to obtain the new hosting decisions. In Line 8, we invoke Algorithms 1 and 2 to get the other control decisions. Lines 9∼10 decide whether a model switch operation occurs. In case of no model switch, then we are in Lines 11∼13 and keep hosting the existing set of models that are being hosted. We still need to update the other control decisions as the inputs could be time-varying. The time complexity of Algorithm 4 is $O(T \cdot N^2 \log(1/\nu))$, based on our previous analysis of the time complexities of Algorithms 1 and 2.

Our motivation for Algorithm 4 is to balance the cumulative switching cost $\sum_t C_{SC}^t$ and the cumulative non-switching cost $\sum_t C_{-SC}^t$, and control this balance by k. We want to intentionally establish the relationship of $\sum_t C_{SC}^t \leq \frac{1}{k} \sum_t C_{-SC}^t$; that is, $\sum_t C_{SC}^t + \sum_t C_{-SC}^t \leq (1 + \frac{1}{k}) \sum_t C_{-SC}^t$. We connect the total cost to the non-switching cost only, and thus remove the concern on the switching cost which used to be hard to manage. Hereafter, we will further connect the non-switching cost to the offline optimum and prove the competitive ratio, as elaborated in our competitive analysis later.

B. Algorithm 5: Hedge-Based Ensemble Learning

As the models are selected and specified by Algorithm 4, the key idea of Algorithm 5 is, for each given data sample, using each model to conduct inference and then combining these inference results as a weighted sum to produce the joint, final inference result that will be returned to the end user. Specifically, we adopt the Hedge algorithm [39] to update the weights in an exponential manner based on the loss between the inferred label of each model and the ground-truth label. Lines 4∼6 conduct weight normalization. Line 8 produces the joint inference result. Lines 11∼13 are the Hedge steps, preparing for the next data sample in the stream. The time complexity of Algorithm 5 is $O(NM^t)$, where the loops in Lines 4 and 11 have N iterations, and Line 2 has M^t iterations.

C. Loss and Competitiveness Analysis

We show that the inference loss cumulatively incurred by our ensemble learning algorithm upon the entire data stream could be no worse than a parameterized constant times the sum of the optimal loss in each time slot incurred by the single best model out of the selected models for that time slot. We also define the competitive ratio and exhibit that our proposed approach leads to a parameterized-constant competitive ratio.

We introduce some additional notations. We write the social cost as $C = \sum_t (C_L^t + C_{-L}^t) = \sum_t (C_L^t + C_{SC}^t + C_{-SC}^t)$, where $C_L^t = \sum_m ((\sum_n \alpha_{n,m}^t h_n^t(a_m^t) + \beta_m^t h^t(a_m^t)) - b_m^t)^2$ is the loss at t, and $C_{-L}^t = C_{SC}^t + C_{-SC}^t$ is the non-loss social cost at t; and C_{SC}^t and C_{-SC}^t are as in Section IV-A.

Theorem 3 *With $\mu_m^t = \sqrt{8 \ln I^t / m}$, we have*

$$\sum_t C_L^t \leq \sum_t C_L^{t*}[1 + \theta_1(2\sqrt{\sum_t \tfrac{M^t}{2} \sum_t \ln I^t} + \sqrt{(\sum_t \ln I^t)\tfrac{T}{8}})].$$

I^t is the number of the models hosted at t. C_L^t is the loss incurred jointly by the I^t models as in Algorithm 5 upon the data samples $\mathcal{M}^t$. C_L^{t*} is the optimal loss incurred by the single best model out of the I^t models upon the data samples $\mathcal{M}^t$, assuming $\sum_t C_L^{t*} \geq \frac{1}{\theta_1} > 0$ where $\theta_1 \neq 0$ is a constant.

Proof. See Appendix C in our supplementary material. □

Theorem 4 *Our entire proposed approach is "r-competitive" for the problem $\mathbb{P}_0$. If we use C to denote the objective value of $\mathbb{P}_0$ incurred by our approach, and C^* to denote the sum of the social cost (except the loss) in the offline optimum of $\mathbb{P}_0$ plus the cumulative per-time-slot optimal loss incurred by the single best model out of the selected models in each corresponding time slot, then we have $C \leq rC^*$, where $r = \max\{r_1, r_2\}$, $r_1 = 1 + \theta_1(2\sqrt{\sum_t \frac{M^t}{2} \sum_t \ln I^t} + \sqrt{(\sum_t \ln I^t)\frac{T}{8}})$, and $r_2 = \rho(1 + \frac{1}{k})(\mathcal{R}^T\theta_2 + 1)$. In r_2, assuming $\sum_t C_{-L}^{t*} \geq \frac{1}{\theta_2} > 0$ where $\theta_2 \neq 0$ is a constant; k is the constant as in Algorithm 4; and ρ and $\mathcal{R}^T$ are also constants.*

Proof. See Appendix D in our supplementary material. □

Theorem 4 captures the competitive ratio and thus the multiplicative gap between the objective value of $\mathbb{P}_0$ incurred by our proposed online approach and that of $\mathbb{P}_0$ in a particularly specified case. This particular case is that we use the social cost (except the inference loss) from the offline optimum of $\mathbb{P}_0$, and use the inference loss incurred cumulatively by the per-time-slot single best model out of the ensemble in each time slot, where the ensemble is chosen as in the offline optimum of $\mathbb{P}_0$. Alternatively, one may want to quantify the multiplicative gap between our approach and the offline optimum of $\mathbb{P}_0$ directly; if so, in the offline optimum of $\mathbb{P}_0$, the inference loss is incurred by the best ensemble which may vary dynamically as time goes. Here, we choose not to directly compare to the offline optimum of $\mathbb{P}_0$ in order to get aligned with the convention of the "learning from experts" setting [39]. The comparison to the offline optimum of $\mathbb{P}_0$ can be of independent interest, which we would like to postpone to future work.

V. Experimental Study

A. Experimental Settings

Datasets: We use two real-world datasets and two widely-used benchmark datasets, all with concept drifts [22]–[26].

- The **Posture** dataset contains $164,860$ data samples from a sensor carried by 5 different people one after another. The task is to predict which movement is performed, among the 11 classes. This dataset was artificially injected with concept drifts according to Reis et. al. [42] in 2016.
- The **Rialto** dataset consists of $82,250$ data samples from time-lapse videos of a webcam and 10 classes [41]. Each data sample is a normalized 27-dimensional RGB histogram (i.e., 27 features), and the 10 classes correspond to 10 colorful buildings next to the Rialto bridge in Venice, Italy in 2016. The task is classification. Concept drifts exist as the changing weather and lighting conditions affect the color-based representation.
- The **Spam** dataset contains $9,324$ email messages (20% spam emails and 80% legitimate emails) derived from the Spam Assassin collection. This data set contains 500 features, which were retrieved using the chi-square feature-selection approach. It has been considered a typical gradual drift dataset since Katakis et. al. [43]. The task is to predict an email is spam or legitimate.
- The **Electricity** dataset contains $45,312$ instances, collected every 30 minutes from the Australian New South Wales Electricity Market between May 1996 and December 1998 [44]. This data set contains 8 features and 2 classes and has been widely used for concept drift adaptation evaluations. The task is to predict the change of the price (i.e., up and down).

Models: We consider $8 \sim 64$ model providers, and divide the first 80% data evenly into $8 \sim 64$ pieces and train the models correspondingly. We use the rest 20% data as the data stream for the service provider, arriving in a uniform manner. Without loss of generality, we adopt the decision trees as the models in our evaluations (unless otherwise specified), aligned with lots of existing ensemble learning research [21], [25], [26]; specifically, we consider the Classification And Regression Tree (CART) [73]. We also consider the Support Vector Machine (SVM) [21], [74] and the Multilayer Perceptron (MLP) (as in deep learning) as other base models for comparison. We consider $|\mathcal{T}| = 100$ minutes, with 1 minute per time slot.

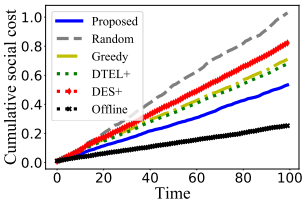

Fig. 5: Social cost

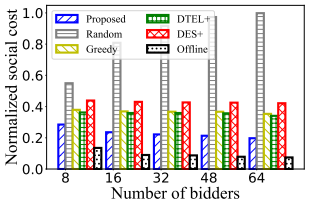

Fig. 6: Scalability

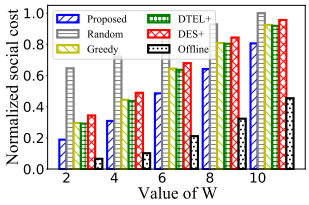

Fig. 7: Impact of hosting budget

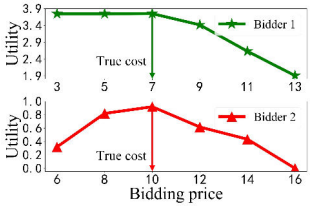

Fig. 8: Utility

Costs and Bids: We consider the Alibaba Cloud [45] of 29 regions and 88 availability zones worldwide [46], with each region often having multiple availability zones. We choose Beijing, China as the location of the service provider's cloud, and use all the other regions with their 76 availability zones as the model providers' clouds (and we consider up to 64 of them), where each availability zone corresponds to a model provider. We can thus use the geographical distance to estimate the network delay as the model downloading cost. We use the electricity cost as the model hosting cost, following the hourly real-time electricity prices of ComEd in January, 2024 [47]. The participation threshold of model providers is uniformly generated in $[0, 1]$ [17], [20], and the service provider's maximum number of models that can be purchased from each auction is in $[6, 10]$. The bidding price is from $[5, 18]$ \$, following the electricity consumption for model training [48]. The model variation indicator is set to 1 (i.e., model unchanged) by default, unless otherwise noted.

Algorithms: We implement multiple algorithms.

- Proposed: Our proposed approach, which uses the Hedge ensemble method to update the weights for the models.
- Random: The approach that selects bids randomly in each time slot, where the model hosting decisions directly follow the bid selection decisions with equal and constant weight for each deployed model.
- Greedy: The approach that optimizes the one-shot slice of our original problem in each individual time slot, i.e., deciding the models to host by solving $\mathbb{P}_3^t$ at each t, uses the same algorithms as our proposed approach for bid and payment determination, and employs equal and constant weight for each hosted model.
- DTEL$^+$: The greedy approach above using the state-of-the-art ensemble learning algorithm DTEL [25] to update the weights for the models. This approach calculates the weight based on the mean squared error, updates the weight for each model only after processing all the data samples in a time slot, and produces the inference result of the ensemble by combining that of each model via a weighted voting scheme.
- DES$^+$: The greedy approach above using the state-of-the-art ensemble learning algorithm with dynamic ensemble selection [22]. For each data sample, this approach finds its nearest neighbors within the set of the data samples in the last time slot, uses the models that can correctly classify these nearest neighbors to construct the ensemble, and produces the inference result of the ensemble via a soft majority voting scheme.
- Offline: The offline optimum of our original problem solved via the Gurobi [75] solver, with all the inputs in the entire time horizon known in advance.

B. Experimental Results

Fig. 5 shows the normalized social cost of different algorithms in the cumulative manner as time goes, when there are 8 bids in every auction. Our approach shows a slower growth trend of the cumulative social cost than all others, and is the closest to the offline optimum. We have also conducted the evaluations on the real-time cumulative social cost for 16, 32, 48, and 64 bids, respectively, and they show similar trends.

Fig. 6 presents the normalized average social cost of different algorithms. As the number of bidders increases, compared to Random, Greedy, DTEL$^+$, and DES$^+$, our approach reduces the social cost on average by 70%, 37%, 35%, and 46%, respectively. Note that even though DES$^+$ has better inference accuracy than DTEL$^+$, the former incurs larger loss when its inference is incorrect, thus greater social cost. The empirical competitive ratio of our approach is $2.1 \sim 2.7$. The social cost of our approach decreases as the number of bidders grows because there are more bids to choose from for further cost reduction, which shows the scalability of our approach.

Fig. 7 shows the impact of the model hosting budget on the normalized average social cost. As more models are allowed to be hosted, the social cost increases for all algorithms; and ours still incurs the lowest social cost, except the offline optimum.

Fig. 8 verifies the truthfulness of our auction mechanism. We select two bids randomly as the example. As shown, if the bid bids its true cost, i.e., 7\$ and 10\$, respectively, then it achieves the highest utility. The utility could drop vastly if the bidding price is not the same as the true cost.

Fig. 9 checks the individual rationality of our auction mechanism. We randomly select two bids and record the changes of their bidding price and corresponding payments over 10 consecutive slots. We see that the payment received is always no less than the bidding price. Note that when a bid does not win in an auction, the payment defaults to zero.

Fig. 10 and Fig. 11 exhibit the dynamic regret and the dynamic fit of different algorithms for $\mathbb{P}_1$, as the total length of the time horizon varies. Our approach has the lowest dynamic regret and fit, also confirming that our regret and fit grow only sub-linearly with time, aligning with our theoretical analysis.

Fig. 12 illustrates how model variations can impact the social cost. In our experiments, we simulate model variations as the model provider using the additional training data from another model provider to update its own model. When models in the market undergo such updates, the social cost exhibits less fluctuation compared to the social cost of models that are static. This is because updated models could more easily adapt to data streams, thereby minimizing the inference loss.

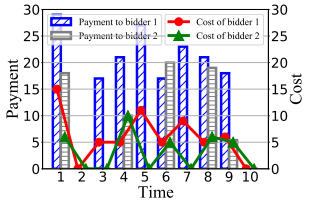

Fig. 9: Payment vs. cost

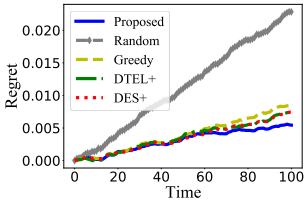

Fig. 10: Dynamic regret

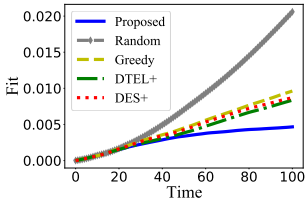

Fig. 11: Dynamic fit

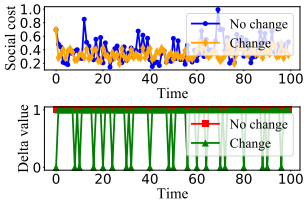

Fig. 12: Impact of model change

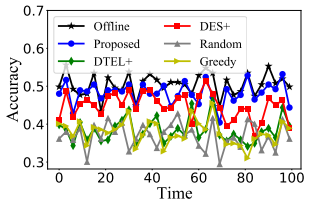

Fig. 13: Real-time accuracy

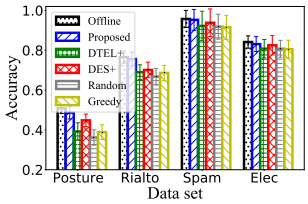

Fig. 14: Accuracy and deviation (I)

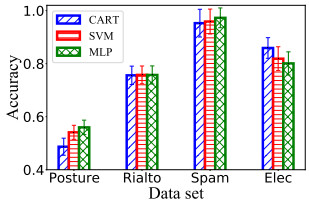

Fig. 15: Accuracy and deviation (II)

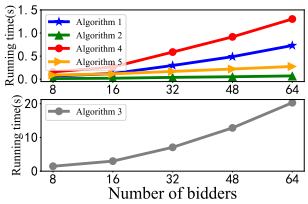

Fig. 16: Algorithm running time

Fig. 13 evaluates the inference accuracy in each time slot over the entire data stream of the Posture data. The results for other datasets are similar. Our approach not only achieves the highest accuracy in all the time slots, but is also relatively more stable; the accuracy results of other algorithms fluctuate dramatically and are more sensitive to the concept drift.

Fig. 14 and Fig. 15 visualize the inference accuracy averaged over time and the standard deviation upon the entire data stream. The standard deviation is calculated based on the inference accuracy of each time slot, representing the sensitivity to concept drifts and the stability of the algorithm [25]. Fig. 14 exhibits significantly better accuracy upon all four data streams and exhibits lower standard deviation compared to others on most data streams, indicating that our approach is less sensitive to concept drifts and is more stable. Fig. 15 exhibits that our algorithmic approach can work with different types of base models in the ensemble learning process. Different base models can be more preferred for different datasets.

Fig. 16 depicts the execution time of each algorithm in our approach. As the number of bidders reaches 64, Algorithms 1$\sim$5 can complete in 0.73 seconds, 0.07 seconds, 20.30 seconds, 1.30 seconds, and 0.28 seconds for each auction, performing well compared to the length of a single time slot. This shows the computational efficiency of our approach in practice.

VI. Discussions

A. Cheap Models with Bad Inference Performance

Our goal in this work is jointly optimizing the cost and the inference loss, as captured by the problem $\mathbb{P}_0$. It is yet crucial to manage the balance or trade-off between the cost and the inference performance of the models. One approach is to associate weights to the different terms in the objective function of $\mathbb{P}_0$. We leave to the service provider to decide the weight for each term in the objective function, depending on its own needs and requirements. In our current evaluations, the weights for the cost of the auctioneer, the cost of the bidders, and the inference loss in the objective are all set equal.

Our strategy for solving $\mathbb{P}_0$ is decoupling the cost from the inference loss, and addressing the former first and the latter afterwards. With this strategy, we can already achieve multiple provable performance guarantees. In particular, Theorem 3 implies that at each time slot, as we update the weights for the models in the ensemble, the inference loss incurred by our model ensemble is no greater than a parameterized constant times the inference loss incurred by the single best model in terms of the inference loss in the ensemble at hindsight.

There could be multiple methods to address the cheap models that have bad inference performance. First, an empirical method could be filtering out such models and ruling out such bids from participating in the auction before the auction starts. To do so, the system should allow the service provider to use a testing dataset to pre-check the performance of each model from the model providers. Second, a more systematic method could be based on modifying our algorithms in the manuscript without filtering out any bids in prior. To simultaneously consider cost and inference performance, we would want to modify our problem $\mathbb{P}_3^t$, solved within Algorithm 4, by adding the inference loss to the objective and adding the constraints (1c)$\sim$(1e). Solving this new $\mathbb{P}_3^t$ faces new challenges: the data samples $\mathcal{M}^t$ are unknown; the new $\mathbb{P}_3^t$ now is a mixed-integer non-linear program, harder to solve; determining the weights for the models here may interfere with our current Algorithm 5; and Algorithm 4 that solves the new $\mathbb{P}_3^t$ can impact our current Theorems 3 and 4. Third, a new and different method could be incorporating the inference loss directly into the process of determining which bids to procure in each auction. This is non-trivial, and will raise the fundamental question of how to ensure truthfulness and individual rationality when part of the objective function for the auction is unknown, i.e., the inference loss is unknown as the data samples are unseen.

B. Duration of Each Time Slot

The time slots correspond to the decision frequency, depending on the system under control and the service provider. Our models, formulations, algorithms, and analysis consider the length of each time slot as pre-specified [76]. Yet, the length of the time slot or the frequency of the control decisions indeed impacts the social cost over time. We consider two cases as follows, both taking exactly the same inputs as $\mathbb{P}_0$.

The first case is that every time slot t is divided into J smaller "time intervals" of equal length, which are indexed as $\mathcal{J} = \{1, 2, ..., J\}$, and we make the control decisions for each time interval $j \in \mathcal{J}$ of each time slot t. The control decisions can thus be written as $x_n^{t,j}$, $y_n^{t,j}$, $z^{t,j}$, $\alpha_{n,m}^{t,j}$, and $\beta_m^{t,j}$. $\mathcal{M}^{t,j}$ can represent the set of data samples that arrive at the time interval j of the time slot t, where $\cup_{j=1}^{J} \mathcal{M}^{t,j} = \mathcal{M}^t$, $\forall t$. Based on these, we can transform the problem $\mathbb{P}_0$ to the problem $\mathbb{P}_0'$. We also consider another problem $\mathbb{P}_0''$, which is defined as $\mathbb{P}_0'$ with the additional constraints that the bid selection and model hosting decisions for all the intervals within a time slot must stay unchanged (but are allowed to change across time slots). For the problems of $\mathbb{P}_0$, $\mathbb{P}_0'$, and $\mathbb{P}_0''$, we use $\mathcal{P}_0$, $\mathcal{P}_0'$, and $\mathcal{P}_0''$ to denote their objective functions, use the subscript OPT to denote the offline optimal objective value, and use $\boldsymbol{X}^*$ to denote the offline optimal solution to $\mathbb{P}_0$. Then, we have

$$\mathcal{P}_{0\,OPT}' \overset{(a)}{\leq} \mathcal{P}_{0\,OPT}'' \overset{(b)}{\leq} \mathcal{P}_0''(\boldsymbol{X}^*) \overset{(c)}{\leq} J \cdot \mathcal{P}_0(\boldsymbol{X}^*) \overset{(d)}{=} J \cdot \mathcal{P}_{0\,OPT}.$$

Inequality (a) holds, as $\mathbb{P}_0'$ and $\mathbb{P}_0''$ have the same objective function, but the latter has a more restricted solution space due to the additional constraints. Inequality (b) holds, because $\mathcal{P}_{0\,OPT}''$ is the optimal value of $\mathbb{P}_0''$. Also, note that $\mathbb{P}_0''$ and $\mathbb{P}_0$ have exactly the same solution space. Inequality (c) holds for any feasible solution, if we just apply the additional constraints to the objective function $\mathcal{P}_0''$. Equality (d) holds, due to the definition of $\boldsymbol{X}^*$. Overall, the offline optimal social cost incurred by such per-time-interval fine-grained control is no greater than J times the offline optimal social cost incurred by the per-time-slot coarse-grained control.

The second case is that the entire time horizon is divided into J larger "time frames" of equal length, which are indexed by $\mathcal{J}$, and each time slot uniquely belongs to a time frame, where the set of all the time slots of the time frame j can be represented as $\mathcal{T}_j$; so, we make the control decisions for each time frame $j \in \mathcal{J}$. Note that in this case the corresponding version of the problem $\mathbb{P}_0'$ can be equivalently defined as $\mathbb{P}_0$ with the additional constraints that the bid selection and model hosting decisions for all the time slots within a time frame must stay unchanged (but are allowed to change across time frames). It is obvious that $\mathcal{P}_{0\,OPT}' \geq \mathcal{P}_{0\,OPT}$, as the former has a more restricted solution space. Overall, the offline optimal social cost incurred by such per-time-frame coarse-grained control is no less than the offline optimal social cost incurred by the per-time-slot fine-grained control.

We have conducted new evaluations to investigate how the length of the time slot can impact the social cost incurred by our proposed approach in practice. For Fig. 17, we take the same inputs as our existing evaluations, with the same length of the entire time horizon, i.e., 100 minutes. Yet, we divide the time horizon into $T = 50$, 100, and 200 time slots, where the length of a single time slot is 2 minutes, 1 minute, and 0.5 minutes, respectively. We see that, as the time slot length increases, the total social cost increases, and that, as the time slot length decreases, the total social cost also decreases.

VII. Related Work

We categorized the previous research into three groups and highlight their drawbacks compared to our work, respectively.

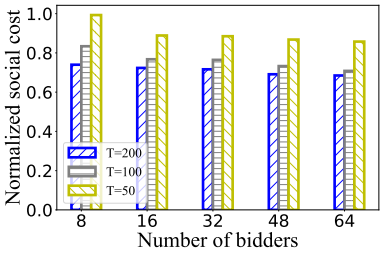

Fig. 17: Social cost under different time slot length

We also summarize and recap the uniqueness of our work.

Ensemble Methods for Concept Drifts: Elwell et. al. [21] trained a new classifier for each batch of data and used a dynamically-weighted majority-voting method to combine these classifiers. Jiao et. al. [22] employed the K-nearest neighbors (KNN) method for dynamically selecting ensemble classifiers for each incoming data sample, and the final classification result is determined through a soft majority vote of multiple base classifiers. Zhao et. al. [23] reused historical models for the ensemble and adaptively adjusted the weights for them. Liu et. al. [24] utilized an instance-weighting method to adjust the sample weights and an ensemble diversity measure to select the classifiers. Sun et. al. [25] exploited a diversity-based strategy to preserve the historical models. Lu et. al. [26] trained individual classifiers by adaptively determining the data chunk size and the classifier weights.

These works study various different ensemble approaches to tackle concept drifts; however, they only consider the inference loss without the system cost and overhead, and none of them has considered the cloud/edge environment.

Ensemble Learning at Cloud/Edge: Yao et. al. [27] built an ensemble of classifiers and combined the prediction results through majority voting in edge-assisted anomaly detection. Stephanie et. al. [28] constructed ensembles via using local models to produce a generalized final model with high accuracy. Shlezinger et. al. [29] proposed edge ensemble which enabling diverse predictors on each device to form a deep ensemble during the inference process. Bai et. al. [30] built a deep neural network ensemble according to the features of admitted inference tasks to optimize the inference quality. Zhang et. al. [31] adopted ensemble methods to improve the classification accuracy for meteorological applications in cloud/edge systems. Zong et. al. [32] employed an ensemble of constituent caching policies for edge caching.

This line of research is often focused on an offline perspective, and does not typically consider the data streaming scenario in an online setting. None of them covers any incentive or market mechanism in the ensemble learning context.

AI Model Markets: Chen et. al. [12] proposed the first formal framework for the ML model marketplace, employing a model-based pricing mechanism instead of data-based pricing. Weng et. al. [33] deployed the marketplace on the blockchain and determined the model price based on the true performance. Sun et. al. [34] adopted a distributed federated-learning-based marketplace with privacy and incentive considerations. Liu et. al. [13] proposed an end-to-end marketplace and built compensation functions for data owners and price functions for buyers. Cao et. al. [35] proposed an edge federated ML

model marketplace with cost evaluation. Nguyen et. al [36] designed a distributed model marketplace based on blockchain and considered the incentive mechanism in IoT networks.

These works mainly adopt direct pricing, rather than auctions, without proving the economic properties as in our work. They also lack the consideration of the long-term constraints and the online setting that feature the problem we study.

Overall, our research in this paper is different from works of the three groups as described above and is superior in the following aspects: (i) our mechanism consists of dynamic and repetitive auctions, capturing the long-term participation of bidders and achieving the desired economic properties in expectation; (ii) our mechanism works for unknown future inputs and streamed data in online settings, while controlling the state switching of model hosting in real time; and (iii) our mechanism adopts the model ensemble method to equip the service with robustness and stability against arbitrary concept drifts in the data streams in the cloud computing environment.

VIII. Conclusion

Provisioning adaptive and accurate inference on streamed data is generally desired in a wide range of cloud services, yet gets largely overlooked in previous research. In this paper, we aim to bridge this gap. We take a unique perspective of acquiring models dynamically from the auction market and combining them with the service provider's own model to conduct ensemble learning. We design online algorithms with provable performance guarantees and economic properties to achieve the long-term optimization of system overhead and inference loss collectively for both the service provider and the model providers. We also conduct comprehensive and thorough evaluations with real-world and benchmark datasets under realistic settings to validate the performance of our approach from a variety of different angles. We hope our work could inspire more new research along this direction.

References

[1] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, "Jointdnn: An efficient training and inference engine for intelligent mobile cloud computing services," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 565–576, 2019.

[2] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[3] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[4] 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, 2018.

[5] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[6] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–36, 2017.

[7] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, pp. 241–258, 2020.

[8] "Modzy." [Online]. Available: https://www.modzy.com/

[9] "Google cloud marketplace." [Online]. Available: https://cloud.google.com/marketplace

[10] "Aws marketplace." [Online]. Available: https://aws.amazon.com/marketplace/

[11] "Azure marketplace." [Online]. Available: https://azuremarketplace.microsoft.com/zh-CN/

[12] L. Chen, P. Koutris, and A. Kumar, "Towards model-based pricing for machine learning in a data marketplace," in *SIGMOD*, 2019.

[13] J. Liu, J. Lou, J. Liu, L. Xiong, J. Pei, and J. Sun, "Dealer: an end-to-end model marketplace with differential privacy," in *VLDB*, 2021.

[14] S. Chen, Z. Zhou, F. Liu, L. I. Zongpeng, and S. Ren, "Cloudheat: An efficient online market mechanism for datacenter heat harvesting," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 3, no. 3, pp. 11.1–11.31, 2018.

[15] S. Chen, L. Jiao, L. Wang, and F. Liu, "An online market mechanism for edge emergency demand response via cloudlet control," in *IEEE INFOCOM*, 2019.

[16] M. Derakhshan, D. M. Pennock, and A. Slivkins, "Beating greedy for approximating reserve prices in multi-unit vcg auctions," in *ACM-SIAM SODA*, 2021.

[17] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *IEEE INFOCOM*, 2015.

[18] Y. Li, F. Li, S. Yang, P. Zhou, L. Zhu, and Y. Wang, "Three-stage stackelberg long-term incentive mechanism and monetization for mobile crowdsensing: An online learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1385–1398, 2021.

[19] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1799–1813, 2019.

[20] Z. Wang, L. Gao, and J. Huang, "Socially-optimal mechanism design for incentivized online learning," in *IEEE INFOCOM*, 2022.

[21] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[22] B. Jiao, Y. Guo, D. Gong, and Q. Chen, "Dynamic ensemble selection for imbalanced data streams with concept drift," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] P. Zhao, L. Cai, and Z. Zhou, "Handling concept drift via model reuse," *Machine learning*, vol. 109, pp. 533–568, 2019.

[24] A. Liu, J. Lu, and G. Zhang, "Diverse instance-weighting ensemble based on region drift disagreement for concept drift adaptation," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 293–307, 2020.

[25] Y. Sun, K. Tang, Z. Zhu, and X. Yao, "Concept drift adaptation by exploiting historical knowledge," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4822–4832, 2018.

[26] Y. Lu, Y. M. Cheung, and Y. Y. Tang, "Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 2764–2778, 2019.

[27] W. Yao, K. Zhang, C. Yu, and H. Zhao, "Exploiting ensemble learning for edge-assisted anomaly detection scheme in e-healthcare system," in *IEEE GLOBECOM*, 2021.

[28] V. Stephanie, I. Khalil, M. S. Rahman, and M. Atiquzzaman, "Privacy-preserving ensemble infused enhanced deep neural network framework for edge cloud convergence," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3763–3773, 2023.

[29] N. Shlezinger, E. Farhan, H. Morgenstern, and Y. C. Eldar, "Collaborative inference via ensembles on the edge," in *IEEE ICASSP*, 2021.

[30] Y. Bai, L. Chen, M. Abdel-Mottaleb, and J. Xu, "Automated ensemble for deep learning inference on edge computing platforms," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4202–4213, 2021.

[31] J. Zhang, P. Liu, F. Zhang, H. Iwabuchi, A. A. d. H. e. A. de Moura, and V. H. C. de Albuquerque, "Ensemble meteorological cloud classification meets internet of dependable and controllable things," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3323–3330, 2021.

[32] T. Zong, C. Li, Y. Lei, G. Li, H. Cao, and Y. Liu, "Cocktail edge caching: Ride dynamic trends of content popularity with ensemble learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 208–219, 2022.

[33] J. Weng, J. Weng, C. Cai, H. Huang, and C. Wang, "Golden grain: Building a secure and decentralized model marketplace for mlaas," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3149–3167, 2021.

[34] P. Sun, X. Chen, G. Liao, and J. Huang, "A profit-maximizing model marketplace with differentially private federated learning," in *IEEE INFOCOM*, 2022.

[35] T.-D. Cao, H.-L. Truong, T. Truong-Huu, and M.-T. Nguyen, "Enabling awareness of quality of training and costs in federated machine learning marketplaces," in *IEEE/ACM UCC*, 2022, pp. 41–50.

[36] L. D. Nguyen, S. R. Pandey, S. Beatriz, A. Broering, and P. Popovski, "A marketplace for trading ai models based on blockchain and incentives for iot data," *arXiv preprint arXiv:2112.02870*, 2021.

[37] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[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] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[40] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic iot management," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1276–1286, 2018.

[41] V. Losing, B. Hammer, and H. Wersing, "Knn classifier with self adjusting memory for heterogeneous concept drift," in *IEEE ICDM*, 2016.

[42] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista, "Fast unsupervised online drift detection using incremental kolmogorov-smirnov test," in *ACM SIGKDD*, 2016.

[43] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, and I. Vlahavas, "An adaptive personalized news dissemination system," *Journal of intelligent information systems*, vol. 32, pp. 191–212, 2009.

[44] M. Harries, N. S. Wales *et al.*, "Splice-2 comparative evaluation: Electricity pricing," 1999.

[45] "Alibaba cloud." [Online]. Available: https://www.aliyun.com/

[46] "Alibaba cloud server geographical and availability zone distribution list." [Online]. Available: https://www.aliyunfuwuqi.com/region/3796/

[47] "Hourly pricing." [Online]. Available: https://hourlypricing.comed.com/live-prices/

[48] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[49] X. Zhao, C. Gu, H. Zhang, X. Yang, X. Liu, J. Tang, and H. Liu, "Dear: Deep reinforcement learning for online advertising impression in recommender systems," in *AAAI*, 2021.

[50] H. Zhu, X. Li, P. Zhang, G. Li, J. He, H. Li, and K. Gai, "Learning tree-based deep model for recommender systems," in *ACM SIGKDD international conference on knowledge discovery & data mining*, 2018.

[51] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[52] F. Orabona, "A modern introduction to online learning," *arXiv preprint arXiv:1912.13213*, 2019.

[53] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[54] X. Wei, H. Yu, and M. J. Neely, "Online primal-dual mirror descent under stochastic constraints," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 2, pp. 1–36, 2020.

[55] S. Balseiro, H. Lu, and V. Mirrokni, "Dual mirror descent for online allocation problems," in *ICML*, 2020.

[56] A. Lobos, P. Grigas, and Z. Wen, "Joint online learning and decision-making via dual mirror descent," in *ICML*, 2021.

[57] N. Eshraghi and B. Liang, "Improving dynamic regret in distributed online mirror descent using primal and dual information," in *LDCC*, 2022.

[58] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2017.

[59] S. Chen, Y.-J. Zhang, W.-W. Tu, P. Zhao, and L. Zhang, "Optimistic online mirror descent for bridging stochastic and adversarial online convex optimization," *Journal of Machine Learning Research*, vol. 25, no. 178, pp. 1–62, 2024.

[60] N. Eshraghi and B. Liang, "Dynamic regret of online mirror descent for relatively smooth convex cost functions," *IEEE Control Systems Letters*, vol. 6, pp. 2395–2400, 2022.

[61] H. Yu and M. J. Neely, "A low complexity algorithm with $o(\sqrt{t})$ regret and $o(1)$ constraint violations for online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 1–24, 2020.

[62] J. Yuan and A. Lamperski, "Online convex optimization for cumulative constraints," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[63] H. Yu, M. Neely, and X. Wei, "Online convex optimization with stochastic constraints," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[64] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2503–2528, 2012.

[65] X. Yi, X. Li, T. Yang, L. Xie, Y. Hong, T. Chai, and K. H. Johansson, "Distributed online convex optimization with adversarial constraints: Reduced cumulative constraint violation bounds under slater's condition," *arXiv preprint arXiv:2306.00149*, 2023.

[66] A. Sinha and R. Vaze, "Playing in the dark: No-regret learning with adversarial constraints," *arXiv preprint arXiv:2310.18955*, 2023.

[67] ——, "Tight bounds for online convex optimization with adversarial constraints," *arXiv preprint arXiv:2405.09296*, 2024.

[68] S. Paternain, S. Lee, M. M. Zavlanos, and A. Ribeiro, "Distributed constrained online learning," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3486–3499, 2020.

[69] S. Mizuno and F. Jarre, "Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation." *Mathematical Programming*, vol. 84, no. 1, 1999.

[70] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.

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

[72] B. H. Korte, J. Vygen, B. Korte, and J. Vygen, *Combinatorial optimization*. Springer, 2011.

[73] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[74] C. Yang, Y. Cheung, J. Ding, and K. C. Tan, "Concept drift-tolerant transfer learning in dynamic environments," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3857–3871, 2021.

[75] "The leader in decision intelligence technology." [Online]. Available: https://www.gurobi.com/

[76] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

Yining Zhang received the B.S. degree in Electronic Information Science and Technology from Shanxi University, China, in 2020. She is currently a Ph.D. student in School of Artificial Intelligence in Beijing University of Posts and Telecommunications. Her research interests are in the areas of online learning, online optimization, and Edge AI.

Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently with the University of Oregon, USA, and was previously a member of technical staff at Nokia Bell Labs, Ireland. He researches AI infrastructures, cloud/edge networks, energy systems, cybersecurity, and multimedia. He has published 70+ papers mainly in leading journals such as IEEE Journal on Selected Areas in Communications, IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, and IEEE Transactions on Parallel and Distributed Systems and conferences such as INFOCOM, MOBIHOC, ICDCS, SECON, and ICNP. He is a U.S. National Science Foundation CAREER awardee, and is also a recipient of the Ripple Faculty Fellowship, the Alcatel-Lucent Bell Labs UK and Ireland Recognition Award, and the Best Paper Awards of IEEE CNS 2019 and IEEE LANMAN 2013. He has been on the program committees as a track chair for ICDCS and as a member for many conferences such as INFOCOM, MOBIHOC, ICDCS, WWW, and IWQoS, and has also served as the program chair of multiple workshops with INFOCOM and ICDCS.

Konglin Zhu received the master's degree in computer Science from the University of California, Los Angeles, CA, USA, and the Ph.D. degree from the University of Göttingen, Germany, in 2009 and 2014, respectively. He is now an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include Internet of Vehicles, Edge Computing and Distributed Learning.

Xiaojun Lin received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 2000 and 2005, respectively. He joined the School of Electrical and Computer Engineering at Purdue University in 2005, and became a Professor of ECE in 2017. Since June 2023, he joined the Department of Information Engineering, The Chinese University of Hong Kong, as a Professor and Global STEM Scholar. Dr. Lin's research interests are in the analysis, control and optimization of large and complex networked systems, including both communication networks and power grid. He received the NSF CAREER award in 2007. He received 2005 best paper of the year award from Journal of Communications and Networks, IEEE INFOCOM 2008 best paper award, ACM MobiHoc 2021 best paper award, and ACM e-Energy 2022 best paper award. He was the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, the TPC co-chair for ACM MobiHoc 2009, the Mini-Conference co-chair for IEEE INFOCOM 2012, and the General cochair for ACM e-Energy 2019. He has served as an Area Editor for (Elsevier) Computer Networks Journal, an Associate Editor for IEEE/ACM Transactions on Networking, and a Guest Editor for (Elsevier) Ad Hoc Networks journal.

Lin Zhang received the B.S. and the Ph.D. degrees in 1996 and 2001, both from the Beijing University of Posts and Telecommunications, Beijing, China. From 2000 to 2004, he was a Postdoctoral Researcher with Information and Communications University, Daejeon, South Korea, and Nanyang Technological University, Singapore, respectively. He joined Beijing University of Posts and Telecommunications in 2004, where he has been a Professor since 2011. He is also the director of Beijing Big Data Center. His current research interests include mobile cloud computing and Internet of Things.

APPENDIX

A. Proof of Theorem 1

Proof. We first define the dynamic regret $\widetilde{\mathrm{Reg}}^T$ and the dynamic fit $\widetilde{\mathrm{Fit}}^T$ for the relaxed problem $\mathbb{P}_2$, and then propose Lemmas 1 and 2 to assist subsequent proofs for the theorem.

$$\widetilde{\mathrm{Reg}}^T = \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^t) - \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^{t*}), \tag{1}$$

$$\widetilde{\mathrm{Fit}}^T = \|[\sum_{t=1}^{T} \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)]^+\|, \widetilde{\boldsymbol{x}}^t \in \widetilde{\mathcal{X}}^t, \forall t. \tag{2}$$

Here, $\widetilde{\boldsymbol{x}}^{t*}$ is the fractional optimal solution, and $\widetilde{\boldsymbol{x}}^{t*} \in \arg\min_{\boldsymbol{x}^t \in \widetilde{\mathcal{X}}^t} f^t(\boldsymbol{x}^t)$, where $\widetilde{\mathcal{X}}^t := \{\boldsymbol{x} | \mathbf{h}^t(\boldsymbol{x}) \succeq \mathbf{0}, d^t(\boldsymbol{x}) \leq 0, \mathbf{g}^t(\boldsymbol{x}) \preceq \mathbf{0}; x_n^t \in [0,1], \forall n\}$.

Our lemmas and theorem reply on some common assumptions [1]–[3] that can be easily satisfied: (1) The function $f^t(\widetilde{\boldsymbol{x}})$ has bounded gradients on $\widetilde{\mathcal{X}}$, i.e., $\|\nabla f^t(\widetilde{\boldsymbol{x}})\| \leq F, \forall \widetilde{\boldsymbol{x}} \in \widetilde{\mathcal{X}}$; and $\mathbf{g}^t(\widetilde{\boldsymbol{x}})$ is bounded on $\widetilde{\mathcal{X}}$, i.e., $\|\mathbf{g}^t(\widetilde{\boldsymbol{x}})\| \leq G, \forall \widetilde{\boldsymbol{x}} \in \widetilde{\mathcal{X}}, \forall t$; (2) The radius of the convex feasible set $\mathcal{X}$ is bounded, i.e., $\|\widetilde{\boldsymbol{x}}_1 - \widetilde{\boldsymbol{x}}_2\| \leq R, \forall \widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2 \in \widetilde{\mathcal{X}}$; (3) There exists a constant $\delta > 0$ and an interior point $\widehat{\boldsymbol{x}} \in \widetilde{\mathcal{X}}$, such that $\mathbf{g}^t(\widehat{\boldsymbol{x}}^t) \leq -\delta \mathbf{1}, \forall t$; (4) The slack constant δ satisfies $\delta > \overline{V}(\mathbf{g})$, where the point-wise maximal variation of the consecutive constraints is denoted as $\overline{V}(\mathbf{g}) = \max_t \max_{\widetilde{\boldsymbol{x}} \in \widetilde{\mathcal{X}}} \|[\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}) - \mathbf{g}^t(\widetilde{\boldsymbol{x}})]^+\|$.

Lemma 1 *The relationship on dynamic regret and fit of $\mathbb{P}_1$ in the domain of integers and reals can be written as*

$$\mathrm{Reg}^T \leq \widetilde{\mathrm{Reg}}^T, \mathrm{Fit}^T \leq \widetilde{\mathrm{Fit}}^T. \tag{3}$$

Proof. We derive the relationship between Reg^T and $\widetilde{\mathrm{Reg}}^T$ as

$$\begin{aligned}
\mathrm{Reg}^T &= E[\sum_{t=1}^{T} f^t(\bar{\boldsymbol{x}}^t)] - \sum_{t=1}^{T} f^t(\boldsymbol{x}^{t*}) \\
&\overset{(a)}{=} \sum_{t=1}^{T} f^t(E[\bar{\boldsymbol{x}}^t]) - \sum_{t=1}^{T} f^t(\boldsymbol{x}^{t*}) \\
&= \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^t) - \sum_{t=1}^{T} f^t(\boldsymbol{x}^{t*}) + \sum_{t=1}^{T} f^t(E[\bar{\boldsymbol{x}}^t]) - \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^t) \\
&\overset{(b)}{\leq} \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^t) - \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^{t*}) = \widetilde{\mathrm{Reg}}^T,
\end{aligned} \tag{4}$$

where (a) holds by the linearity of $f^t(\bar{\boldsymbol{x}}^t)$, (b) holds due to $E[\bar{\boldsymbol{x}}^t] = \widetilde{\boldsymbol{x}}^t$ which ensured by randomized rounding algorithm and the fact that the objective value conducted by integer optimum is more than fractional optimum. We derive fit as

$$\begin{aligned}
\mathrm{Fit}^T &= \|[E[\textstyle\sum_{t=1}^{T} \mathbf{g}^t(\bar{\boldsymbol{x}}^t)]]^+\| \overset{(a)}{\leq} \|E[\textstyle\sum_{t=1}^{T} \mathbf{g}^t(\bar{\boldsymbol{x}}^t)]\| \\
&\overset{(b)}{=} \|\textstyle\sum_{t=1}^{T} \mathbf{g}^t(E[\bar{\boldsymbol{x}}^t])\| = \|\textstyle\sum_{t=1}^{T} \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\| = \widetilde{\mathrm{Fit}}^T,
\end{aligned} \tag{5}$$

where (a) follows that the value of the 2-norm will decrease due to all the negative values are set to 0 by using $[\cdot]^+ = \max\{\cdot, 0\}$. The linearity of $\mathbf{g}^t(\bar{\boldsymbol{x}}^t)$ and the unchanged expectation property holds by the randomized rounding algorithm guarantee (b).

□

Lemma 2 *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda}^1 = \mathbf{0}$, we have*

$$\tfrac{(\|\boldsymbol{\lambda}^{t+1}\|^2 - \|\boldsymbol{\lambda}^t\|^2)}{2} \leq \eta \boldsymbol{\lambda}^t \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t) + \tfrac{\eta^2}{2} \|\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\|^2, \tag{6}$$

$$\|\boldsymbol{\lambda}^t\| \leq \|\overline{\boldsymbol{\lambda}}\| := \eta G + \tfrac{2FR + R^2/(2\gamma) + (\eta G^2)/2}{\delta - \overline{V}(\mathbf{g})}, \forall t. \tag{7}$$

Proof. According to the update policy of $\boldsymbol{\lambda}$, we have

$$\begin{aligned}
\|\boldsymbol{\lambda}^{t+1}\|^2 &= \|[\boldsymbol{\lambda}^t + \eta \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)]^+\|^2 \leq \|\boldsymbol{\lambda}^t + \eta \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\|^2 \\
&= \|\boldsymbol{\lambda}^t\|^2 + 2\eta \boldsymbol{\lambda}^t \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t) + \eta^2 \|\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\|^2.
\end{aligned} \tag{8}$$

After rearranging terms in (8), we obtain

$$\tfrac{(\|\boldsymbol{\lambda}^{t+1}\|^2 - \|\boldsymbol{\lambda}^t\|^2)}{2} \leq \eta \boldsymbol{\lambda}^t \mathbf{g}^t(\widetilde{\boldsymbol{x}}^t) + \tfrac{\eta^2}{2} \|\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\|^2. \tag{9}$$

Since $\widetilde{\boldsymbol{x}}^{t+1}$ is the optimum for objective in (6) of our main paper, by using the interior point $\widehat{\boldsymbol{x}}^t$ mentioned in assumption 3, we have

$$\begin{aligned}
&\nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1} \mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) + \tfrac{\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma} \\
&\leq \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1} \mathbf{g}^t(\widehat{\boldsymbol{x}}^t) + \tfrac{\|\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma} \\
&\overset{(a)}{\leq} \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t) - \delta \boldsymbol{\lambda}^{t+1} \mathbf{1} + \tfrac{\|\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma} \\
&\overset{(b)}{\leq} \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t) - \delta \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{\|\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma},
\end{aligned} \tag{10}$$

where inequality (a) holds due to assumption 2, and inequality (b) holds because $\|\boldsymbol{\lambda}^{t+1}\|$ is less or equal to $\boldsymbol{\lambda}^{t+1}\mathbf{1}$ for any non-negative vector $\boldsymbol{\lambda}^{t+1}$. Then we have

$$\begin{aligned}
&\boldsymbol{\lambda}^{t+1} g^t(\widetilde{\boldsymbol{x}}^{t+1}) \\
&\leq \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t) - \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t) \\
&-\delta \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{(\|\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|^2)}{2\gamma} \\
&\overset{(a)}{\leq} \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t) - \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t) - \delta \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{R^2}{2\gamma} \\
&\overset{(b)}{\leq} \|\nabla f^t(\widetilde{\boldsymbol{x}}^t)\|(\|\widehat{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^t\| + \|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|) - \delta \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{R^2}{2\gamma} \\
&\overset{(c)}{\leq} 2FR - \delta \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{R^2}{2\gamma} \overset{def}{=} \Phi^{t+1},
\end{aligned} \tag{11}$$

where inequality (a) holds by the bounded radius on the domain, and $\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\| \geq 0$; inequality (b) holds by Cauchy-Schwartz inequality; the inequality (c) holds by using the bounded gradient in assumption 1 and bounded domain. We consider (11) with lemma 2, and have

$$\begin{aligned}
&\Delta(\boldsymbol{\lambda}^{t+1}) := \tfrac{(\|\boldsymbol{\lambda}^{t+1}\|^2 - \|\boldsymbol{\lambda}^t\|^2)}{2} \\
&\leq \eta \boldsymbol{\lambda}^{t+1} \mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) + \tfrac{\eta^2}{2} \|\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1})\|^2 \\
&\overset{(a)}{\leq} \eta \boldsymbol{\lambda}^{t+1}(\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) - \mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1})) + \tfrac{\eta^2 G^2}{2} + \Phi^{t+1} \\
&\overset{(b)}{\leq} \eta \boldsymbol{\lambda}^{t+1}[\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) - \mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1})]^+ + \tfrac{\eta^2 G^2}{2} + \Phi^{t+1} \\
&\overset{(c)}{\leq} \eta \overline{V}(\mathbf{g}) \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{\eta^2 G^2}{2} + 2FR - \delta \|\boldsymbol{\lambda}^{t+1}\| + \tfrac{R^2}{2\gamma},
\end{aligned} \tag{12}$$

where inequality (a) holds by using the upper-bound of g; inequality (b) holds by the non-negative property of $\boldsymbol{\lambda}^{t+1}$; and inequality (c) holds by assumption 4.

Next, we show the correctness of (7) by contradiction. Without loss of generality, we suppose that $t+2$ is the first time index that breaks (7), namely:

$$\|\boldsymbol{\lambda}^{t+1}\| \leq \|\overline{\boldsymbol{\lambda}}\| < \|\boldsymbol{\lambda}^{t+2}\|. \tag{13}$$

By using the update policy of $\boldsymbol{\lambda}$, the relationship can be obtained on $\boldsymbol{\lambda}$ between consecutive time slots as follows:

$$\begin{aligned}
&\|\boldsymbol{\lambda}^{t+1}\| \overset{(a)}{\geq} \|\boldsymbol{\lambda}^{t+2}\| - \|\boldsymbol{\lambda}^{t+2} - \boldsymbol{\lambda}^{t+1}\| \\
&= \|\boldsymbol{\lambda}^{t+2}\| - \|[\boldsymbol{\lambda}^{t+1} + \eta \mathbf{g}^{t+1}(\boldsymbol{x}^{t+1})]^{+} - \boldsymbol{\lambda}^{t+1}\| \\
&\geq \|\boldsymbol{\lambda}^{t+2}\| - \|\boldsymbol{\lambda}^{t+1} + \eta \mathbf{g}^{t+1}(\boldsymbol{x}^{t+1}) - \boldsymbol{\lambda}^{t+1}\| \\
&= \|\boldsymbol{\lambda}^{t+2}\| - \|\eta \mathbf{g}^{t+1}(\boldsymbol{x}^{t+1})\| \overset{(b)}{>} \|\overline{\boldsymbol{\lambda}}\| - \eta G, \qquad (14)
\end{aligned}$$

where inequality (a) holds by the triangle inequality, and inequality (b) holds by (13). Then we can obtain $\Delta(\boldsymbol{\lambda}^{t+1}) < 0$, leading to $\|\boldsymbol{\lambda}^{t+2}\| < \|\boldsymbol{\lambda}^{t+1}\|$, which contradicts $\|\boldsymbol{\lambda}^{t+1}\| \leq \|\overline{\boldsymbol{\lambda}}\| < \|\boldsymbol{\lambda}^{t+2}\|$. Thus, Lemma 2 can be proved.

□

Here, we begin the proof of Theorem 1.

Step I: The objective in (6) of our main paper implies that it is $1/\gamma$-strongly convex with respect to $\widetilde{\boldsymbol{x}}$, denoted by $J^t(\widetilde{\boldsymbol{x}}^t)$, i.e., $\forall \widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2 \in \widetilde{\mathcal{X}}$:

$$J^t(\widetilde{\boldsymbol{x}}_2) \geq J^t(\widetilde{\boldsymbol{x}}_1) + \nabla J^t(\widetilde{\boldsymbol{x}}_1)(\widetilde{\boldsymbol{x}}_2 - \widetilde{\boldsymbol{x}}_1) + \frac{\|\widetilde{\boldsymbol{x}}_2 - \widetilde{\boldsymbol{x}}_1\|^2}{2\gamma}. \quad (15)$$

Since $\widetilde{\boldsymbol{x}}^{t+1}$ is the optimum for $\min_{\widetilde{\boldsymbol{x}} \in \widetilde{\mathcal{X}}} J^t(\widetilde{\boldsymbol{x}})$, then we have

$$\nabla J^t(\widetilde{\boldsymbol{x}}^{t+1})(\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}) \geq 0. \quad (16)$$

We set $\widetilde{\boldsymbol{x}}_1 = \widetilde{\boldsymbol{x}}^{t+1}$ and $\widetilde{\boldsymbol{x}}_2 = \widetilde{\boldsymbol{x}}^{t*}$, and plugging (16) into (15), we have

$$J^t(\widetilde{\boldsymbol{x}}^{t*}) \geq J^t(\widetilde{\boldsymbol{x}}^{t+1}) + \tfrac{1}{2\gamma}\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2. \quad (17)$$

After adding $f^t(\widetilde{\boldsymbol{x}}^t)$ on both two sides, expanding $J^t(\cdot)$ according to its definition and using the property of convex function on $f^t(\cdot)$, i.e., $f^t(\widetilde{\boldsymbol{x}}^{t*}) \geq f^t(\widetilde{\boldsymbol{x}}^t) + \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t)$, we have

$$\begin{aligned}
&f^t(\widetilde{\boldsymbol{x}}^t) + \nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) + \tfrac{\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma} \\
&\leq f^t(\widetilde{\boldsymbol{x}}^{t*}) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma} - \tfrac{\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2}{2\gamma} \\
&\overset{(a)}{\leq} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2}{2\gamma} - \tfrac{\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2}{2\gamma}, \qquad (18)
\end{aligned}$$

where inequality (a) holds by $\boldsymbol{\lambda}^{t+1} \succeq \mathbf{0}$ and the per-slot optimal solution $\widetilde{\boldsymbol{x}}^{t*}$ is feasible, i.e., $\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t*}) \preceq \mathbf{0}$. Then we analyze the gradient term as

$$\begin{aligned}
&-\nabla f^t(\widetilde{\boldsymbol{x}}^t)(\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t) \leq \|\nabla f^t(\widetilde{\boldsymbol{x}}^t)\|\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\| \\
&\overset{(a)}{\leq} \tfrac{\|\nabla f^t(\widetilde{\boldsymbol{x}}^t)\|^2}{2\zeta} + \tfrac{\zeta}{2}\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|^2 \overset{(b)}{\leq} \tfrac{F^2}{2\zeta} + \tfrac{\zeta}{2}\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|^2, \qquad (19)
\end{aligned}$$

where ζ is an arbitrary positive constant. Inequality (a) holds by $a^2 + b^2 \geq 2ab$, inequality (b) holds by the bounded gradient of f^t. Then we plug (19) into (18) and get

$$\begin{aligned}
&f^t(\widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) \leq f^t(\widetilde{\boldsymbol{x}}^{t*}) + (\tfrac{\zeta}{2} - \tfrac{1}{2\gamma})\|\widetilde{\boldsymbol{x}}^{t+1} - \widetilde{\boldsymbol{x}}^t\|^2 \\
&+\tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2) + \tfrac{F^2}{2\zeta} \\
&\overset{(a)}{=} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2) + \tfrac{\gamma F^2}{2}, \qquad (20)
\end{aligned}$$

where inequality (a) holds by setting $\zeta = 1/\gamma$. By plugging (20) into the Lemma 2, we have

$$\begin{aligned}
&\tfrac{\Delta(\boldsymbol{\lambda}^{t+1})}{\eta} + f^t(\widetilde{\boldsymbol{x}}^t) \leq \boldsymbol{\lambda}^{t+1}\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) + \tfrac{\eta}{2}\|\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1})\|^2 \\
&+f^t(\widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) - \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) \\
&= f^t(\widetilde{\boldsymbol{x}}^t) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) + \tfrac{\eta}{2}\|\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1})\|^2 \\
&+\boldsymbol{\lambda}^{t+1}\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) - \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1}) \\
&\overset{(a)}{\leq} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2) + \tfrac{\gamma F^2}{2} \\
&+\tfrac{\eta}{2}\|\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1})\|^2 + \boldsymbol{\lambda}^{t+1}(\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) - \mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1})) \\
&\leq f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2) + \tfrac{\gamma F^2}{2} \\
&+\tfrac{\eta G^2}{2} + \boldsymbol{\lambda}^{t+1}[\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}^{t+1}) - \mathbf{g}^t(\widetilde{\boldsymbol{x}}^{t+1})]^{+} \\
&\overset{(b)}{\leq} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2) + \tfrac{\gamma F^2}{2} \\
&+\tfrac{\eta G^2}{2} + \|\boldsymbol{\lambda}^{t+1}\|V(\mathbf{g}^t), \qquad (21)
\end{aligned}$$

where inequality (a) holds by using (20), inequality (b) holds by assumption 4. Then we consider the intermediate terms as follows:

$$\begin{aligned}
&\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 = \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^t\|^2 - \|\widetilde{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^{t-1*}\|^2 + \|\widetilde{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^{t-1*}\|^2 \\
&\overset{(a)}{=} \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t-1*}\|\|\widetilde{\boldsymbol{x}}^{t*} - 2\widetilde{\boldsymbol{x}}^t + \widetilde{\boldsymbol{x}}^{t-1*}\| + \|\widetilde{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^{t-1*}\|^2 \\
&\overset{(b)}{\leq} 2R\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t-1*}\| + \|\widetilde{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^{t-1*}\|^2, \qquad (22)
\end{aligned}$$

where equation (a) holds by applying difference of two squares on the first two terms, inequality (b) holds by the triangle inequality for vectors and the bounded radius on domain. After applying (22) to (21), we have

$$\begin{aligned}
&\tfrac{\Delta(\boldsymbol{\lambda}^{t+1})}{\eta} + f^t(\widetilde{\boldsymbol{x}}^t) \leq f^t(\widetilde{\boldsymbol{x}}^{t*}) + \|\boldsymbol{\lambda}^{t+1}\|V(\mathbf{g}^t) + \tfrac{\gamma F^2}{2} + \tfrac{\eta G^2}{2} \\
&+\tfrac{1}{2\gamma}(2R\|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t-1*}\| + \|\widetilde{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^{t-1*}\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2), \qquad (23)
\end{aligned}$$

Sum over t, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \tfrac{\Delta(\boldsymbol{\lambda}^{t+1})}{\eta} + \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^t) \leq \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{\gamma F^2 T}{2} + \tfrac{\eta G^2 T}{2} \\
&+\tfrac{R \cdot V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{\gamma} + \sum_{t=1}^{T}\{\|\boldsymbol{\lambda}^{t+1}\|V(\mathbf{g}^t)\} \\
&+\tfrac{1}{2\gamma}\sum_{t=1}^{T}(\|\widetilde{\boldsymbol{x}}^t - \widetilde{\boldsymbol{x}}^{t-1*}\|^2 - \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t+1}\|^2) \\
&\overset{(a)}{\leq} \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{\gamma F^2 T}{2} + \tfrac{\eta G^2 T}{2} + \tfrac{R \cdot V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{\gamma} \\
&+\|\overline{\boldsymbol{\lambda}}\|\sum_{t=1}^{T} V(\mathbf{g}^t) + \tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^1 - \widetilde{\boldsymbol{x}}^{0*}\|^2 - \|\widetilde{\boldsymbol{x}}^{T*} - \widetilde{\boldsymbol{x}}^{T+1}\|^2) \\
&\overset{(b)}{\leq} \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^{t*}) + \tfrac{\gamma F^2 T}{2} + \tfrac{\eta G^2 T}{2} + \tfrac{R \cdot V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{\gamma} \\
&+\|\overline{\boldsymbol{\lambda}}\|V(\{\mathbf{g}^t\}_{t=1}^T) + \tfrac{1}{2\gamma}(\|\widetilde{\boldsymbol{x}}^1 - \widetilde{\boldsymbol{x}}^{0*}\|^2), \qquad (24)
\end{aligned}$$

where $V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T) := \sum_{t=1}^T \|\widetilde{\boldsymbol{x}}^{t*} - \widetilde{\boldsymbol{x}}^{t-1*}\|$, inequality (a) holds by the definition of $\|\overline{\boldsymbol{\lambda}}\|$, inequality (b) holds due to $V(\{\mathbf{g}^t\}_{t=1}^T) := \sum_{t=1}^T \max_{\widetilde{\boldsymbol{x}} \in \widetilde{\mathcal{X}}} \|[\mathbf{g}^{t+1}(\widetilde{\boldsymbol{x}}) - \mathbf{g}^t(\widetilde{\boldsymbol{x}})]^{+}\|$. Then,

$$\begin{aligned}
&\widetilde{\mathrm{Reg}}^T = \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^t) - \sum_{t=1}^{T} f^t(\widetilde{\boldsymbol{x}}^{t*}) \leq \tfrac{\gamma F^2 T}{2} + \|\overline{\boldsymbol{\lambda}}\|V(\{\mathbf{g}^t\}_{t=1}^T) \\
&+\tfrac{\eta G^2 T}{2} + \tfrac{R \cdot V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{\gamma} + \tfrac{(\|\widetilde{\boldsymbol{x}}^1 - \widetilde{\boldsymbol{x}}^{0*}\|^2)}{2\gamma} - \sum_{t=1}^{T} \tfrac{\Delta(\boldsymbol{\lambda}^{t+1})}{\eta} \\
&= \tfrac{\gamma F^2 T}{2} + \|\overline{\boldsymbol{\lambda}}\|V(\{\mathbf{g}^t\}_{t=1}^T) + \tfrac{\eta G^2 T}{2} + \tfrac{R \cdot V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{\gamma}
\end{aligned}$$

$+\frac{(\|\widetilde{\boldsymbol{x}}^1-\widetilde{\boldsymbol{x}}^{0*}\|^2)}{2\gamma}-\frac{\|\boldsymbol{\lambda}^{T+2}\|^2}{2\eta}+\frac{\|\boldsymbol{\lambda}^{t=2}\|^2}{2\eta}\overset{(a)}{\leq}\mathcal{R}^T,$ (25)

where $\mathcal{R}^T=\frac{R\cdot V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{\gamma}+\frac{\gamma F^2T}{2}+\frac{\eta G^2(T+1)}{2}+\frac{R^2}{2\gamma}+\|\overline{\boldsymbol{\lambda}}\|V(\{\mathbf{g}^t\}_{t=1}^T)$, inequality (a) holds due to $\|\widetilde{\boldsymbol{x}}^1-\widetilde{\boldsymbol{x}}^{0*}\|^2$ has been bounded by R according to bounded radius of domain, $\|\boldsymbol{\lambda}^{T+2}\|^2\geq 0$, and $\|\boldsymbol{\lambda}^{t=2}\|^2\leq\eta^2G^2$ if $\boldsymbol{\lambda}^1=\mathbf{0}$.

Step II: According to the dual recursion in Algorithm 1, we have:

$$[\boldsymbol{\lambda}^T+\eta\mathbf{g}^T(\widetilde{\boldsymbol{x}}^T)]^+\geq\ldots\geq\boldsymbol{\lambda}^1+\sum_{t=1}^T\eta\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t).\quad(26)$$

Since $\boldsymbol{\lambda}^1=\mathbf{0}$, we can rearrange the terms, and then obtain

$$\sum_{t=1}^T\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\leq\frac{\boldsymbol{\lambda}^{T+1}}{\eta}-\frac{\boldsymbol{\lambda}^1}{\eta}\leq\frac{\boldsymbol{\lambda}^{T+1}}{\eta}.\quad(27)$$

Therefore, $\widetilde{\mathrm{Fit}}^T=\|[\sum_{t=1}^T\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)]^+\|$ can be treated as

$$\widetilde{\mathrm{Fit}}^T\leq\|\textstyle\sum_{t=1}^T\mathbf{g}^t(\widetilde{\boldsymbol{x}}^t)\|\leq\|\frac{\boldsymbol{\lambda}^{T+1}}{\eta}\|\leq\frac{\|\overline{\boldsymbol{\lambda}}\|}{\eta}.\quad(28)$$

Then we can obtain the following results: $\mathrm{Reg}^T\leq\widetilde{\mathrm{Reg}}^T\leq\mathcal{R}^T$ and $\mathrm{Fit}^{\mathrm{T}}\leq\widetilde{\mathrm{Fit}}^{\mathrm{T}}\leq\frac{\boldsymbol{\lambda}^{T+1}}{\eta}\leq\frac{\|\overline{\boldsymbol{\lambda}}\|}{\eta}$. By choosing proper step sizes, we can express these bounds as sub-linear functions of T. The dynamic regret and the dynamic fit can be bounded by controlling step sizes as $\gamma=\eta=\max\{\sqrt{\frac{V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)}{T}},\sqrt{\frac{V(\{\mathbf{g}^t\}_{t=1}^T)}{T}}\}$, then we have $\mathrm{Reg}^T=\mathcal{O}(\max\{\sqrt{V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)T},\sqrt{V(\{\mathbf{g}^t\}_{t=1}^T)T}\})$ and $\mathrm{Fit}^{\mathrm{T}}\leq\frac{\|\overline{\lambda}\|}{\eta}=\mathcal{O}(\max\{\frac{T}{V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)},\frac{T}{V(\{\mathbf{g}^t\}_{t=1}^T)}\})$. If we further set $\gamma=\eta=\mathcal{O}(T^{-\frac{1}{3}})$, we can get $\mathrm{Reg}^T=\mathcal{O}(\max\{V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)T^{\frac{1}{3}},V(\{\mathbf{g}^t\}_{t=1}^T)T^{\frac{1}{3}},T^{\frac{2}{3}}\})$ and $\mathrm{Fit}^{\mathrm{T}}=\mathcal{O}(T^{\frac{2}{3}})$. Then the sub-linear regret and fit of $\mathcal{O}(T^{\frac{2}{3}})$ can be achieved if we have $V(\{\widetilde{\boldsymbol{x}}^{t*}\}_{t=1}^T)\in\boldsymbol{o}(T^{\frac{2}{3}})$ and $V(\{\mathbf{g}^t\}_{t=1}^T)\in\boldsymbol{o}(T^{\frac{2}{3}})$. □

B. Proof of Theorem 2

Step I: We aim to prove that $E(\bar{x}_n^t)$ is monotonically non-increasing in c_n^t. We make $C(\bar{x}_n^t,c_n^t,\boldsymbol{c}_{-n}^t)$ denote the objective value of $\mathbb{P}_1$ with reported prices $(c_n^t,\boldsymbol{c}_{-n}^t)$. where c_n^t denotes the bidding price of model provider n and $\boldsymbol{c}_{-n}^t$ denotes all the other prices except n. We define $\widetilde{x}_n^t$ and $\dot{\widetilde{x}}_n^t$ as the optimal fractional results of n with bid c_n^t and $\dot{c}_n^t$ with fixing $\boldsymbol{c}_{-n}^t$. In the case that $c_n^t\geq\dot{c}_n^t$, we have $C(\widetilde{x}_n^t,c_n^t,\boldsymbol{c}_{-n}^t)\leq C(\dot{\widetilde{x}}_n^t,c_n^t,\boldsymbol{c}_{-n}^t)$ and $C(\dot{\widetilde{x}}_n^t,\dot{c}_n^t,\boldsymbol{c}_{-n}^t)\leq C(\widetilde{x}_n^t,\dot{c}_n^t,\boldsymbol{c}_{-n}^t)$. By combining the above inequalities together, we have $\widetilde{x}_n^tc_n^t+\dot{\widetilde{x}}_n^t\dot{c}_n^t\leq\dot{\widetilde{x}}_n^tc_n^t+\widetilde{x}_n^t\dot{c}_n^t,\Rightarrow\widetilde{x}_n^t(c_n^t-\dot{c}_n^t)\leq\dot{\widetilde{x}}_n^t(c_n^t-\dot{c}_n^t)\Rightarrow\widetilde{x}_n^t\leq\dot{\widetilde{x}}_n^t,\Rightarrow E(\bar{x}_n^t)\leq E(\dot{\bar{x}}_n^t)$.

Step II: We denote $\chi_n^t=\frac{\widetilde{x}_n^t}{\gamma}+2\lambda_n^{t+1}$ as the upper bound of the integral of $\int_0^\infty E(\bar{x}_n^t)dc$. Based on our statement in Section III-C, we know $\int_0^{\chi_n^t}\widetilde{x}_n^t(c,\boldsymbol{c}_{-n}^t)dc<\infty$ and $\int_{\chi_n^t}^\infty\widetilde{x}_n^t(c,\boldsymbol{c}_{-n}^t)dc=0$. Then we have $\int_0^\infty E(\bar{x}_n^t(c,\boldsymbol{c}_{-n}^t))dc=\int_0^{\chi_n^t}E(\bar{x}_n^t(c,\boldsymbol{c}_{-n}^t))dc+\int_{\chi_n^t}^\infty E(\bar{x}_n^t(c,\boldsymbol{c}_{-n}^t))dc<\infty$.

Step III: For the individual rationality in expectation, we have $p_n^t=c_n^tE(\bar{x}_n^t(c_n^t,\boldsymbol{c}_{-n}^t))+\int_{c_n^t}^{\chi_n^t}E(\bar{x}_n^t(c,\boldsymbol{c}_{-n}^t))dc$ and $U_n^t=p_n^t-c_n^tE(\bar{x}_n^t(c_n^t,\boldsymbol{c}_{-n}^t))=\int_{c_n^t}^{\chi_n^t}E(\bar{x}_n^t(c,\boldsymbol{c}_{-n}^t))dc\geq 0$.

C. Proof of Theorem 3

Lemma 3 *(Theorem 2.3 in [4]) Assume the loss function l is convex in its first argument and takes values in $[0,1]$. For all $M^t\geq 1$, the regret of the exponentially weighted average forecaster with time-varying parameter $\mu_m^t=\sqrt{8\ln I^t/m}$ satisfies*

$$C_L^t-C_L^{t*}\leq 2\sqrt{\tfrac{M^t}{2}\ln I^t}+\sqrt{\tfrac{\ln I^t}{8}}.$$

The proof is based on a reduction from our scenario to standard exponentially weighted average forecaster. We let the hosted set $\{\mathcal{I}\,|y_n^t=1,z^t=1,\forall n,t\}$ be the expert pool, and denote the number of models hosted at t as I^t and the length of the data sequence at t as M^t. Then we plug the expert number $N=I^t$ and the number of instances $n=M^t$ into Theorem 2.3 in [4]. We use j_m to lower bound the weight $ln\frac{w_{j_m,m}^t}{W_m^t}$ by keeping track of the currently best expert, where $w_{i,m}^t$ represents the weight of the hosted model i, $W_m^t=\sum_{i=1}^{I^t}w_{i,m}^t=1$; and j_m is the index of the expert with the smallest loss after the first m rounds.

Then we have $\frac{1}{\mu_m^t}\ln\frac{w_{j_{m-1},m-1}^t}{W_{m-1}^t}-\frac{1}{\mu_{m+1}^t}\ln\frac{w_{j_m,m}^t}{W_m^t}=(\frac{1}{\mu_{m+1}^t}-\frac{1}{\mu_m^t})\ln\frac{W_m^t}{w_{j_m,m}^t}+\frac{1}{\mu_m^t}\ln\frac{\dot{w}_{j_m,m}^t/\dot{W}_m^t}{w_{j_m,m}^t/W_m^t}+\frac{1}{\mu_m^t}\ln\frac{w_{j_{m-1},m-1}^t/W_{m-1}^t}{\dot{w}_{j_m,m}^t/\dot{W}_m^t}$. We now bound separately the three terms on the right-hand side and get $(\frac{1}{\mu_{m+1}^t}-\frac{1}{\mu_m^t})\ln\frac{W_m^t}{w_{j_m,m}^t}\leq(\frac{1}{\mu_{m+1}^t}-\frac{1}{\mu_m^t})\ln I^t$, $\frac{1}{\mu_m^t}\ln\frac{\dot{w}_{j_m,m}^t/\dot{W}_m^t}{w_{j_m,m}^t/W_m^t}\leq(\frac{1}{\mu_{m+1}^t}-\frac{1}{\mu_m^t})\ln I^t$, and $\frac{1}{\mu_m^t}\ln\frac{w_{j_{m-1},m-1}^t/W_{m-1}^t}{\dot{w}_{j_m,m}^t/\dot{W}_m^t}\leq C_{L,j_m,m}^t-C_{L,j_{m-1},m-1}^t-l(\hat{b}_m^t,b_m^t)+\frac{\mu_m^t}{8}$. We substitute back in the main equation the bounds on the three terms, then we obtain $l(\hat{b}_m^t,b_m^t)\leq C_{L,j_m,m}^t-C_{L,j_{m-1},m-1}^t+\frac{\sqrt{\sigma\ln I^t}}{8\sqrt{m}}+2(\frac{1}{\mu_{m+1}^t}-\frac{1}{\mu_m^t})\ln I^t+\frac{1}{\mu_{m+1}^t}\ln\frac{w_{j_m,m}^t}{W_m^t}-\frac{1}{\mu_m^t}\ln\frac{w_{j_{m-1},m-1}^t}{W_{m-1}^t}$. Summing over m, we have $C_L^t\leq C_L^{t*}+\frac{\sqrt{\sigma M^t\ln I^t}}{4}+2\sqrt{\frac{(M^t+1)\ln I^t}{\sigma}}-\sqrt{\frac{\ln I^t}{\sigma}}$. By choosing $\sigma=8$ to trade off the two main terms, we get $C_L^t\leq C_L^{t*}+2\sqrt{\frac{M^t}{2}\ln I^t}+\sqrt{\frac{\ln I^t}{8}}$.

Finally we perform the global loss performance analysis on the whole data stream by summing over t, and get $\sum_tC_L^t\leq\sum_tC_L^{t*}+\sum_t(2\sqrt{\frac{M^t}{2}\ln I^t}+\sqrt{\frac{\ln I^t}{8}})\overset{(a)}{\leq}\sum_tC_L^{t*}+2\sqrt{\sum_t\frac{M^t}{2}\sum_t\ln I^t}+\sqrt{(\sum_t\ln I^t)\frac{T}{8}}\leq\sum_tC_L^{t*}[1+\theta_1(2\sqrt{\sum_t\frac{M^t}{2}\sum_t\ln I^t}+\sqrt{(\sum_t\ln I^t)\frac{T}{8}})]=\sum_tC_L^{t*}r_1$, where the inequality (a) holds by applying Cauchy-Schwarz inequality.

D. Proof of Theorem 4

According to Algorithm 4, we denote the time slots recorded by $\hat{t}$ as $\{\hat{t}_1,\hat{t}_2,\ldots\}$. Consider any $\vartheta\geq 1$, for the consecutive time slots $\{\hat{t}_\vartheta,\hat{t}_\vartheta+1,\ldots,\hat{t}_{\vartheta+1}-1\}$, we have $C_{-SC}^{\hat{t}_\vartheta}(\bar{\boldsymbol{y}}^{\hat{t}_\vartheta},\bar{\boldsymbol{y}}^{\hat{t}_{\vartheta-1}})\leq\frac{1}{k}\sum_{\tau=\hat{t}_\vartheta}^{\hat{t}_{\vartheta+1}-1}C_{-SC}^\tau(\bar{\boldsymbol{x}}^\tau,\bar{\boldsymbol{y}}^\tau,\bar{z}^\tau)$ due to Line 5 of Algorithm 4. Both sides sum over ϑ, we have $\sum_tC_{SC}^t\leq\frac{1}{k}\sum_tC_{-SC}^t$. Then we can obtain $\sum_tC_{-L}^t=\sum_tC_{SC}^t+\sum_tC_{-SC}^t\leq(1+\frac{1}{k})\sum_tC_{-SC}^t$.

We denote $\rho = \max_t \frac{\max_{\boldsymbol{y}^t,z^t} C^t_{-SC}(\boldsymbol{y}^t,z^t)}{\min_{\boldsymbol{y}^t,z^t} C^t_{-SC}(\boldsymbol{y}^t,z^t)}$, and get $\sum_t f^t(\boldsymbol{x}^t) - \sum_t f^t(\boldsymbol{x}^{t*}) \leq \mathcal{R}^T$ according to the Theorem 1. Then, we have $\sum_t C^t_{-SC} \leq \sum_t \max_{\boldsymbol{y}^t,z^t} C^t_{-SC}(\boldsymbol{y}^t,z^t) \leq \rho \sum_t \min_{\boldsymbol{y}^t,z^t} C^t_{-SC}(\boldsymbol{y}^t,z^t) = \rho(\sum_t f^t(\boldsymbol{x}^t)+\{\sum_t \sum_n y^t_n(v^t_n + u^t_n(1-\Delta^t_n)) + \sum_t e^t z^t\}^*) \leq \rho(\mathcal{R}^T + \sum_t f^t(\boldsymbol{x}^{t*}) + \{\sum_t \sum_n y^t_n(v^t_n + u^t_n(1-\Delta^t_n)) + \sum_t e^t z^t\}^*) = \rho(\mathcal{R}^T + \sum_t C^{t*}_{-SC}) \leq \rho(\mathcal{R}^T + \sum_t C^{t*}_{-L})$.

Based on all the above, we can obtain $\sum_t C^t_{-L} = \sum_t C^t_{SC} + \sum_t C^t_{-SC} \leq (1+\frac{1}{k}) \sum_t C^t_{-SC} \leq \rho(1+\frac{1}{k})(\mathcal{R}^T + \sum_t C^{t*}_{-L}) \leq \rho(1+\frac{1}{k})(\mathcal{R}^T\theta_2 + 1) \sum_t C^{t*}_{-L}$. Jointly with Theorem 3, we can prove that our approach is r-competitive for $\mathbb{P}_0$, where $r = \max\{r_1, r_2\}$, $r_1 = 1 + \theta_1(2\sqrt{\sum_t \frac{M^t}{2} \sum_t \ln I^t} + \sqrt{(\sum_t \ln I^t)\frac{T}{8}})$, and $r_2 = \rho(1+\frac{1}{k})(\mathcal{R}^T\theta_2 + 1)$.

REFERENCES

[1] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[2] 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.

[3] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic iot management," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1276–1286, 2018.

[4] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.