Performance Evaluation 152 (2021) 102249

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Competitive bidding strategies for online linear optimization with inventory management constraints

Russell Lee [a,*], Yutao Zhou [a], Lin Yang [a], Mohammad Hajiesmaili [a], Ramesh Sitaraman [a,b]

[a] University of Massachusetts Amherst, Amherst, MA, USA
[b] Akamai Technologies, Cambridge, MA, USA

ARTICLE INFO

Article history:
Available online 6 October 2021

Keywords:
Online bidding strategy design
Inventory management
Energy storage systems
Electricity market

ABSTRACT

This paper develops competitive bidding strategies for an online linear optimization problem with inventory management constraints in both cost minimization and profit maximization settings. In the minimization problem, a decision maker should satisfy its time-varying demand by either purchasing units of an asset from the market or producing them from a local inventory with limited capacity. In the maximization problem, a decision maker has a time-varying supply of an asset that may be sold to the market or stored in the inventory to be sold later. In both settings, the market price is unknown in each timeslot and the decision maker can submit a finite number of bids to buy/sell the asset. Once all bids have been submitted, the market price clears and the amount bought/sold is determined based on the clearing price and submitted bids. From this setup, the decision maker must minimize/maximize their cost/profit in the market, while also devising a bidding strategy in the face of an unknown clearing price. We propose DEMBID and SUPBID, two competitive bidding strategies for these online linear optimization problems with inventory management constraints for the minimization and maximization setting respectively. We then analyze the competitive ratios of the proposed algorithms and show that the performance of our algorithms approaches the best possible competitive ratio as the maximum number of bids increases. As a case study, we use energy data traces from Akamai data centers, renewable outputs from NREL, and energy prices from NYISO to show the effectiveness of our bidding strategies in the context of energy storage management for a large energy customer participating in a real-time electricity market.

1. Introduction

In this paper, we study online linear optimization problems with inventory management constraints in a bidding scenario. In each time slot, a decision maker must satisfy an online arrival of asset demand $d(t)$ in the cost minimization setting, while it receives an online supply of asset $u(t)$ in the profit maximization setting. The asset may be bought at a cost or sold at a profit with the online arrival of price $p(t)$, where the buying amount $x(t)$ or selling amount $y(t)$ must also be decided online. With inventory management, the decision maker may utilize an inventory of capacity B to store

* Corresponding author.
E-mail addresses: rclee@cs.umass.edu (R. Lee), yutaozhou@cs.umass.edu (Y. Zhou), linyang@cs.umass.edu (L. Yang), hajiesmaili@cs.umass.edu (M. Hajiesmaili), ramesh@cs.umass.edu (R. Sitaraman).

https://doi.org/10.1016/j.peva.2021.102249
0166-5316/

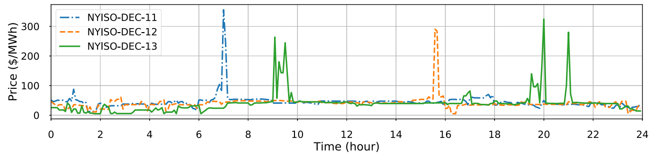


Fig. 1. The energy price dynamics in the NYISO market in three consecutive days in December 2018.

the asset between time slots. With bidding strategies, the price $p(t)$ and decision variables $x(t)$, $y(t)$ are not immediately known in the current time slot — the decision maker must first submit a set of bids on the asset, after which $p(t)$, $x(t)$, and $y(t)$ are made known based on the submitted bids. The problems are considered ProfitMax and CostMin for the maximization and minimization versions.

Online linear optimization is well-studied and recent work on the inventory management variant has also provided worst-case optimality guarantees for both maximization and minimization versions [1,2]. The bidding setting, however, introduces new challenges in online algorithm design because the online price inputs are not even known in the current time slot. Specifically, in most prior work in online algorithms, even though the future input is assumed to be unknown, the online input for the current slot is known accurately. In this paper, we consider a case that the online input for the current slot is also unknown. In such scenario, the online decision maker participates in a market-based bidding mechanism to determine the amount of asset that should be traded (sold or bought) in the market.

From the technical perspective, the unknown price exacerbates the challenges of online algorithm design such that prior algorithms that are designed for the settings with known current price are not applicable for the bidding scenarios. Specifically, prior online algorithms are developed based on carefully-designed threshold functions that determine the trading amount based on the current utilization of the inventory and the known current price [1,2]. In the bidding scenario with unknown price, it is not possible to directly apply those threshold functions anymore and hence it is needed to design strategies for both variants of the problem, i.e., ProfitMax and CostMin.

In addition to introducing new technical challenges, both ProfitMax and CostMin are of significant practical relevance for the timely problem of bidding strategy design for participation of energy storage systems in real-time electricity markets. The bidding strategy for storage participation in markets is different from traditional energy resources since energy storage systems are flexible resources that pose unique physical and operational characteristics. The first unique challenge is the *flexibility in the generation:* energy storage systems are flexible in releasing energy, i.e., it is possible to store energy and release it at future times if that is more beneficial. Traditionally, the trade in the electricity market was based on the fact that the energy could not be stored, and the market operations and bidding strategies were designed based on this fundamental assumption.

The second challenge is *uncertainty:* energy storage systems often provide energy alongside variable renewable generation, which results in inherent uncertainty in their energy generation. In addition, the energy market is highly uncertain and market prices change dynamically based on supply and demand fluctuations. A sample trajectory of real-time energy prices in NYISO is demonstrated in Fig. 1, where one can see irregular and unpredictable patterns in three different days.

The above two challenges make bidding strategy design for storage-assisted parties a fundamentally different problem than the traditional energy market participants. However, the problem of interest in this paper can fully capture both challenges: the flexibility in generation could be captured by the inventory management constraints, and the uncertainty could be capture by the online algorithm design framework.

1.1. Our contributions

In this paper, we tackle the bidding strategy design problems for CostMin and ProfitMax using a principled approach grounded on online algorithm design [3]. This framework enables designing bidding strategies that are provably robust against uncertainty. This paper makes the following contributions:

Algorithm design. We develop two online algorithms and analyze their performance using *competitive ratio* as a well-established metric for online algorithms. The competitive ratio is defined as maximum ratio between an *offline* optimal algorithm with full information on inputs and the limited information online algorithm. The design of our bidding strategies is inspired by prior storage management algorithms in a simplified setting in which the market price is known in advance [1,2,4]. However, in bidding strategy design problems, the decision maker submits bids without knowing the market price. This may result in declining the bid, jeopardizing the feasibility of the online solution. Furthermore, the actual amount of asset traded will also be uncertain, which introduces the challenge of underbidding and overbidding on the asset due to the unknown price. Hence, the existing algorithms [1,2,4] are not applicable to the bidding scenario. We utilize the possibility of submitting multiple bids [5] to the market and resolve these challenges for bidding strategies for both demand and supply sides.

Table 1
Summary of key notations.

Inputs	
T	The number of time slots, $T \geq 1$
$\mathcal{T}$	Set $\mathcal{T} = \{1, 2, \dots, T\}$
$p(t)$	Market clearing price at t, $p_{\min} \leq p(t) \leq p_{\max}$
θ	The price fluctuation ratio, i.e., $\theta = p_{\max}/p_{\min}$
B	The capacity of storage system
ρ_c	Charge rate limit of storage system
ρ_d	Discharge rate limit of storage system
$b(t)$	The storage level (state of charge) at the end of t
Notations for CostMin	
$d(t)$	The asset demand at t
$x(t)$	**opt. variable** The buying quantity at t
Notations for ProfitMax	
$u(t)$	The asset generation at t
$y(t)$	**opt. variable** The selling quantity at t
Notations for bidding	
m	The maximum number of possible bids
$< p_i, q_i >$	Bid i including bidding price p_i and bidding quantity q_i
$\mathcal{B}$	The set of bids, i.e., $\mathcal{B} = \{< p_1, q_1 >, \dots, < p_m, q_m >\}$

Competitive analysis. As the theoretical contribution, we characterize the competitive ratio of both proposed algorithms as a function of number of bids, and show the competitive ratios approaches those values of the basic algorithms [1,2] as the number of bids grows. More specifically, let α be the optimal competitive ratio of the basic algorithm proposed in [2] for CostMin. Our competitive analysis shows that the proposed bidding strategy for CostMin achieves the competitive ratio of $\alpha(\theta/\alpha)^{1/(m-1)}$, where m is the maximum number of bids. In addition, our bidding strategy for ProfitMax achieves the competitive ratio of $(\ln\theta+1)\theta^{1/(m-1)}$, showing a degradation factor of $\theta^{1/(m-1)}$ as compared to the optimal competitive ratio of $(\ln\theta + 1)$ for the basic algorithm with known current price proposed in [1]. For both algorithms as $m \to \infty$, the degradation factors due to unknown prices vanishes and the competitive ratios approaches to those optimal values for the basic settings with known prices.

Empirical evaluations. Lastly, we empirically evaluate our bidding algorithms using extensive data traces of electricity prices from NYISO [6], energy demands from Akamai data centers [7], and renewable production values from solar [8] and wind [9] generation. In an extensive set of experiments, the performance of our algorithms is only 5% worse than the cases in which the price of the market for the incoming slot is known in advance. In addition, our algorithms outperform alternative baseline algorithms by more than 10%, on average. Finally, our results show that as the number of available bids increases, the performance of our algorithms approach the ideal performance of algorithms which know the price in advance.

2. Problem formulation

In this section, we present the system model and cast two optimization problems: CostMin that formulates the cost minimization problem for a storage-assisted customer (StrDEM) participating in the asset market and ProfitMax, a profit maximization problem that focuses on storage-assisted supplier (StrSUP). While the formulated problems in this section are generic and could be applied to any bidding scenarios with inventory management constraints, we present the problems in the context of storage participation in the electricity market (see Table 1).

2.1. System model

We consider a time-slotted model, such that the time horizon T is divided into multiple slots with equal length, e.g., 5 min in CAISO and NYISO [10], each of which is indexed by t. Shortly before slot t, StrDEM (and/or StrSUP) along with other participants submit their bids for the next slot. The clearing price $p(t)$ is determined shortly after the participants submit their bids. Hence, the market participants do not know the value of $p(t)$ for the incoming slot. We assume that the minimum and maximum possible values of the clearing price over the time horizon are known in advance, i.e., $p_{\min} \leq p(t) \leq p_{\max}$. This assumption is reasonable since these values could be predicted from the historical data of prices. We denote parameter θ as the ratio between the maximum and the minimum clearing prices, i.e., $\theta = p_{\max}/p_{\min}$.

For both StrDEM and StrSUP, we assume that there is a single storage system that could be a single large-scale physical storage or a collection of multiple small storage units aggregated as a single virtual storage. Let B, ρ_c, and ρ_d be the capacity, the maximum charge rate, and the maximum discharge rate of the storage system. Furthermore, we denote $b(t) \in [0, B]$ as the storage level at the end of slot t.

2.2. Cost minimization problem

We consider the following scenario. At each timeslot, demand $d(t)$ arrives online that must be satisfied by either submitting bids to the market with the unknown market clearing price $p(t)$, or drawing from the storage system. The decision maker can purchase extra assets from the market to keep in storage for future use. The goal is to design a bidding algorithm to determine the value of $x(t)$ as the procurement amount in each slot, so that the aggregate long-term cost of purchases is minimized and the demand is satisfied.

Storage Model. By denoting $b(t) \in [0, B]$ as the storage level at the end of slot t, the evolution of storage level is given by

$$b(t) = b(t-1) + x(t) - d(t),$$

In addition, we have

$$0 \leq x(t) \leq d(t) + \min\{\rho_c, B - b(t-1)\},$$

that limits the procurement to the demand plus the minimum between the charge rate ρ_c and the available capacity. Finally,

$$x(t) \geq d(t) - \min\{\rho_d, b(t-1)\},$$

limits the procurement to the demand minus the minimum between the maximum discharge rate ρ_d and the available energy in the storage.

Problem Formulation. If the $d(t)$ and $p(t)$ are known for the entire time horizon in advance, the *offline* version of cost minimization problem for StrDEM, can be formulated as a linear program as follows.

$$\begin{aligned}
\textsc{CostMin}: \quad \min \quad & \sum_{t \in \mathcal{T}} p(t)x(t) \\
\text{s.t.}: \quad & \forall t \in \mathcal{T}: \\
& b(t) = b(t-1) + x(t) - d(t), && (1) \\
& x(t) \geq d(t) - \min\big\{\rho_d, b(t-1)\big\}, && (2) \\
& x(t) \leq d(t) + \min\big\{\rho_c, B - b(t-1)\big\}, && (3) \\
& 0 \leq b(t) \leq B, && (4) \\
\text{vars.}: \quad & \{x(t), b(t)\} \in \mathbb{R}^+.
\end{aligned}$$

The objective is to minimize the cost of buying for the market. Constraint (1) dictates the evolution of the storage, and constraints (2)–(4) enforce the capacity and rate limits of the storage.

2.3. Profit maximization problem

There is a storage-assisted supplier (StrSUP) that produces some amount of the asset in each timeslot. In the example of energy storage, this would be electricity from renewable sources such as wind farm or solar plant. The StrSUP has on-site storage systems to store the asset mainly for mitigating the uncertainty of asset generation, and secondly, for strategic supply to the market with potentially higher price to maximize the profit.

Asset Output. We denote the asset output of StrSUP at slot t by $u(t) \geq 0$. We assume that the value of $u(t)$ is known for the incoming slot when submitting the bid. However, the future values of $u(t)$ are not known in advance. Depending on the bidding strategy, $u(t)$ could be directly committed to the market, partially be committed to the market and the residual is stored on the storage, or entirely be stored on the storage for future supply to the market with higher price.

Given the asset output $u(t)$, the goal is to design a bidding strategy to determine the value of $y(t)$, i.e., the supply quantity in each slot, such that the long-term profit of selling the asset to the market with time-varying clearing price is maximized.

Storage Model. Given the asset output $u(t)$ and the supply quantity $y(t)$, the evolution of the storage is given by

$$b(t) = b(t-1) - y(t) + u(t),$$

In addition, we have

$$0 \leq y(t) \leq u(t) + \min\Big\{\rho_d, b(t-1)\Big\},$$

that limits the maximum supply in each slot by the discharge rate limit ρ_d. Also, we have

$$y(t) \geq u(t) - \min\{\rho_c, B - b(t-1)\},$$

that dictates the minimum supply amount to the market given the current storage level and the charging rate limit ρ_c of the storage.

Profit Maximization Problem. The objective is to maximize the cumulative profit obtained by StrSUP over the time horizon. The profit maximization problem ProfitMax is formulated as

$$\begin{aligned}
\textsc{ProfitMax}: \quad \max \quad & \sum_{t\in\mathcal{T}} p(t)y(t) \\
\text{s.t.}: \quad & \forall t \in \mathcal{T}: \\
& 0 \le b(t) \le B, && (5)\\
& y(t) \le u(t) + \min\{\rho_d, b(t-1)\}, && (6)\\
& y(t) \ge u(t) - \min\{\rho_c, B - b(t-1)\}, && (7)\\
& b(t) = b(t-1) - y(t) + u(t), && (8)\\
\text{vars.}: \quad & \{y(t), b(t)\} \in \mathbb{R}^+,
\end{aligned}$$

where the constraints follow from the properties of the storage and the asset availability.

2.4. Tackling online problems

Both CostMin and ProfitMax are linear programs that could be solved effectively in an offline manner [11,12]. However, in practice both problems should be solved in online scenario with the price $p(t)$ as the common online variable for both problems, and the asset demand $d(t)$ as the additional online variable for CostMin, and the asset output $u(t)$ as the online variable for ProfitMax. Hence, we are interested in developing *online algorithms* for both problems that make decisions in each time t, knowing the past and current prices and demands/outputs, but not knowing those same inputs for the future.

Designing online algorithms that implicitly dictate the asset storage charging/discharging decisions, however, is challenging since in an online setting, we do not know if such decisions will work out favorably in the future. In the literature, both problems have been tackled in online scenarios using different analytical approaches. In prior work [2,4], there is a simplifying assumption that the price $p(t)$ is known for the incoming slot. Using competitive analysis [3], online algorithms have been proposed for both simplified problems. The goal is to design algorithms with the smallest *competitive ratio*, that is, the cost ratio between the online algorithm and an *offline* optimal algorithm that has access to *complete* input sequence. Our bidding strategies in this paper will be built on top of these existing competitive online algorithms.

The Structure of Bids. Tackling CostMin and ProfitMax in the bidding scenario introduces two key challenges. First, even the price for the current slot is not known to the decision maker. In other words, since the price will be cleared after gathering all bids from suppliers and customers, bidding strategies should submit their bids without knowing $p(t)$ for the current slot. Second, the amount of asset traded to the market also faces uncertainty, which must be mitigated by an appropriate bidding strategy.

In this setting, the bidding strategy design refers to the way that StrDEM or StrSUP determine its bids. A bid includes:

▷ **Bidding price** denoted as $\hat{p}(t) \in [p_{\min}, p_{\max}]$, i.e., for the supplier, the minimum price at which StrSUP desires to supply the asset to the market; and for the customer, the maximum price at which StrDEM is willing to buy from the market.

▷ **Bidding quantity** denoted as $q(t) \ge 0$, i.e., the amount of asset at which StrDEM (resp. StrSUP) buys from (resp. supplies to) the market at slot t given the bid is accepted.[1]

In the example of electricity markets, usually participants can submit multiple bids with different bidding prices and quantities. For example, in the PJM market, each participant can submit at most 10 bids [5]. To capture this, let m be the maximum number of bids for each participant. Then, we denote $\langle \hat{p}_i(t), q_i(t)\rangle$ as the ith bid in slot t. And we let $\mathcal{B} = \{\langle p_1(t), q_1(t)\rangle, \ldots, \langle p_m(t), q_m(t)\rangle\}$ be the set of submitted bids at slot t. Without loss of generality, we assume the supply bids are indexed in ascending order based on their bidding price, i.e., $p_1(t) \le p_2(t) \le \cdots \le p_m(t)$.

Once the market is cleared and $p(t)$ is known, at the supply side, the bids with the bidding price less than or equal to $p(t)$ are accepted to supply their quantity into the market. More specifically, for StrSUP, let $0 \le k \le m$ be the index of the last accepted bid. Hence, StrSUP will sell $y(t)$ to the market in time t based on k bids:

$$y(t) = \sum_{i=1}^{k} q_i(t). \tag{9}$$

Similarly, for StrDEM, the bids with the bidding price greater than or equal to $p(t)$ will get accepted to buy their quantity from the market. For demand bids, we assume that the bids are indexed in descending order of their bidding price, i.e., $p_1(t) \ge p_2(t) \ge \cdots \ge p_m(t)$. And letting k as the index of the last accepted bid, we have

$$x(t) = \sum_{i=1}^{k} q_i(t). \tag{10}$$

[1] We assume that the market is big enough to absorb/satisfy the bidding quantities of StrDEM and StrSUP entirely and the participants satisfy the minimum requirement for participation.

Given the above relationship between the supply variable $y(t)$ in ProfitMax, the procurement variable $x(t)$ in CostMin, and the bidding quantities in Eqs. (9) and (10), the goal of the bidding strategy design in this work is to set the bids so as to maximize the long-term profit of StrSUP and respectively minimize the long-term cost of StrDEM. Note that the variables $x(t)$ and $y(t)$ are characterized by the index k, and therefore take up to $m+1$ different values. Since the it is impossible to know which of these values $x(t)$ or $y(t)$ will take, one challenge in bidding strategy design is ensuring all $m+1$ possible outcomes are equally favorable. In the next section, we present the details of the proposed bidding strategies.

3. Online bidding strategy design

In this section, we develop our bidding strategies for CostMin, called DEMBID in Section 3.1, and for ProfitMax, called SUPBID, in Section 3.3. In the case where the market price is known *a priori*, the high-level goal of online algorithms for both StrDEM and StrSUP is to store more of the asset when the market price is cheap and discharge more from storage when the price is high. With known market prices for the current slot, DEM-ON (ARP in [2]) and SUP-ON (sOffer in [4]) achieve the best possible competitive ratio among online algorithms. The main ideas are to utilize carefully designed functions that determine reservation prices based on the storage level, and to introduce the notion of virtual storage for tackling the uncertainty of demand and supply.

However, the market price is not known for the incoming slot in the bidding scenario. Hence, the above online algorithms cannot be directly applied. The main challenges are as follows: (1) without knowing the market price, it is not possible to directly use DEM-ON and SUP-ON since their required input of current price is unknown; (2) the possibility of failure of bids may lead to the case that the demand is not fully covered in CostMin, and the waste of renewable output in ProfitMax if there is not enough room in storage to store; and (3) the number of successfully accepted bids is uncertain, and any number of accepted bids should result in a desirable amount of traded asset.

The high-level ideas of our bidding strategy design for addressing the above challenges are as follows. *First*, to tackle the challenge of unknown of market price, we advocate the idea of submitting multiple bids with different bidding prices. For each bid, the bidding quantity is designed in compliance with an adaptive reservation strategy. In this way, we can mimic the adaptive reservation policy to reserve the asset with controllable deviations. *Second,* to tackle the challenge of possibility of infeasible solution due to failure of the bids, we follow the idea of dividing the bids into two categories: The first category is devoted to make sure that we respect the constraints of the problem. Specifically, in each slot, for CostMin, we reserve a *deterministic* bid whose bidding price is set to $p_{\max}$ and the bidding quantity is the minimum required asset to prevent the infeasible solution. That is, if the current maximum discharge possibility of storage, i.e., $\min\{\rho_d, b(t-1)\}$, is less than the demand $d(t)$, the algorithm will submit a deterministic bid with the bidding price of $p_{\max}$ and bidding quantity of $d(t)-\min\{\rho_d, b(t-1)\}$. *Third,* the remaining bids will be designed by applying the so-called *adaptive reservation policy* in [2] and a carefully designed partitioning strategy proposed in this paper. The bidding prices and quantities are partitioned to provide performance guarantees regardless of the market price and which bids are accepted. For ProfitMax we follow the same design paradigm of submitting a deterministic bid and using a partitioning strategy for the remaining bids. In the following, we develop two algorithms following these high-level ideas. Note that for simplicity, we present our algorithms without taking into account the rate constraints of the storage systems. Specifically, we focus on the case where the storage has only capacity constraints, but no rate constraints, i.e., $\rho_c = \rho_d = B$. However, our algorithms could be extended to include such constraints by projecting the values $x(t)$ or $y(t)$ into the feasible region of respecting those constraints. This is done by restricting the bidding quantities such that $x(t)$ respects the upper and lower bounds from constraints (2)–(4), and similarly $y(t)$ for constraints (6)–(7).

3.1. Bidding strategy design for StrDEM

We summarize the pseudocode of DEMBID as Algorithm 1. The main ideas of DEMBID are: (i) the construction of virtual storage, which tackles the challenges due to the demand dynamics; (ii) adaptive reservation based on storage level, which tackles the challenges due to price dynamics in the future; and (iii) multiple-bid strategy, which mitigates the risk of unknown market price and failed bids. The first two are based on prior work [2], and the third one is the novelty of this work. We discuss the details in the following.

Virtual Storage. Inspired from ideas in [2] for designing an online algorithm with the optimal competitive ratio, DEMBID deals with demand dynamics by defining a notion of virtual storage. DEMBID views the demand in each time slot as flexible demand that must be purchased from the market with some degree of freedom obtained by shifting it using storage. To utilize this opportunity, DEMBID constructs several virtual storages to record the satisfied amount of the demand from the market. Specifically, in each time slot with demand $d(t) > 0$, DEMBID initiates an additional virtual storage whose capacity is equal to $d(t)$. (Lines 31–35). The first virtual storage is always representative of the actual physical storage, with capacity B. (Lines 1–3). Finally, when $b(t) = 0$, DEMBID has just used up the physical inventory to satisfy demand. With no leftover asset to assist with future demand, the previous virtual storages have no further use. The number of virtual storages is reset to 1, renewing the process as in Line 28. On the other hand, if $b(t) > 0$, there is leftover asset that can help meet future demand. Then the status of the virtual storages will be not be renewed.

Adaptive Reservation Price. The storage will be regulated by an *adaptive reservation policy*, which deals with price dynamics by defining a notion of *reservation price*. Having a properly constructed reservation price, DEMBID charges the

Algorithm 1 DEMBID: The bidding algorithm for demand-side participant, for each $t \in \mathcal{T}$.

1: // Initialization at $t = 1$
2: $v \leftarrow 1$;
3: $B_1 \leftarrow B$;
4: $\xi_1 \leftarrow [p_{\max}/\alpha]$;
  // The main algorithm for each t
5: $\mathcal{B} \leftarrow \emptyset$
6: $m(t) \leftarrow m$
  // Deterministic bid: To ensure covering demand
7: **if** $d(t) > b(t-1)$ **then**
8:   $x_{\min} \leftarrow d(t) - b(t-1)$
9:   $\mathcal{B} \leftarrow \mathcal{B} \cup < p_{\max}, x_{\min} >$
10:   $m(t) \leftarrow m(t) - 1$
11: **end if**
  // Decide bidding prices
12: $r_d \leftarrow (\theta/\alpha)^{1/m(t)}$
13: $p_0 \leftarrow p_{max}/\alpha$
14: **for** each $i = 1, 2, \dots, m(t)$ **do**
15:   $p_i \leftarrow \frac{p_0}{r_d^i}$
16: **end for**
  // Decide bidding quantities
17: **for** each $j = 1, 2, \dots, m(t)$ **do**
18:   $q_i \leftarrow \sum_{j=1}^{v} \left[G_{B_j}(p_i) - G_{B_j}(\min(p_{i-1}, \xi_v)\right]^+$
19: **end for**
20: $\mathcal{B} \leftarrow \mathcal{B} \cup \left\{< p_1, q_1 >, < p_2, q_2 >, \dots, < p_{m(t)}, q_{m(t)} >\right\}$
21: Submit $\mathcal{B}$
  // Receive $p(t)$ and update the physical storage
22: $k \leftarrow$ index of last accepted bid
23: $x(t) \leftarrow \sum_{i=1}^{k} q_i$
24: $b(t) \leftarrow b(t-1) + x(t) - d(t)$
25: **for** each $i = 1, 2, \dots, v$ **do**
26:   $\xi_i \leftarrow \min\{\xi_i, p(t)\}$
27: **end for**
  // Update the virtual storage
28: **if** $b(t) = 0$ **then**
29:   Initialize the algorithm by executing Lines (2)-(4)
30: **else**
31:   **if** $d(t) > 0$ **then**
32:     $v \leftarrow v + 1$
33:     $B_v \leftarrow d(t)$
34:     $\xi_v \leftarrow p_{\max}/\alpha$
35:   **end if**
36: **end if**

storage if the bidding price is cheaper than the reservation price; otherwise, it discharges from the storage. DEMBID adaptively determines the reservation prices based on the available storage level and updates its value whenever the storage level is changed (see Lines 25–27). DEMBID determines the reservation price for the jth virtual inventory with capacity B_j by the following function,

$$G_{B_j}(p) = \alpha B_j \ln\left[\left(1 - \frac{p}{p_{\max}}\right)\frac{\alpha}{\alpha - 1}\right],\ p \in \left[p_{\min}, \frac{p_{\max}}{\alpha}\right], \tag{11}$$

where

$$\alpha = \left(W\left(-\frac{\theta - 1}{\theta \exp(1)}\right) + 1\right)^{-1}, \tag{12}$$

and W denotes *Lambert-W function* defined as inverse of $f(z) = z\exp(z)$, and $\theta = p_{\max}/p_{\min}$ is the price fluctuation ratio.

Note that this function is carefully designed to guarantee a worst-case competitive ratio for the problem. Further, in [2], it is shown that with this reservation function, DEM-ON achieves the best possible competitive ratio of α as defined in Eq. (12). In particular, the choice of α is motivated by being the exact same competitive ratio as the k-min search problem [13] in the case of $k \to \infty$. In this problem, the buyer seeks to buy $k \geq 1$ units of an asset to minimize their cost. At each timeslot $t = \{1, \ldots, T\}$, the buyer is presented price $p(t)$, and must immediately decide whether to buy some of the asset at the given price. In fact, the problem of online linear optimization with inventory management is a generalization of k-min search when the market price is known.

Multiple-bid Strategy. The cost of the bidding strategy is ultimately determined by the last accepted bid $p_k(t)$. With an asymptotically large amount of bids, DEMBID has sufficient available bids to closely estimate $p_k(t) \approx p(t)$. So, the bidding strategy of DEMBID should converge to optimal online DEM-ON. With this in mind, we design DEMBID to submit bidding quantities q_i that correspond with the adaptive threshold of DEM-ON.

DEMBID first sets a *deterministic* bid to ensure satisfying the demand (see Lines 7–11), while the remaining bids are set according to an adaptive reservation policy. The function G_{B_j} represents the target amount of stored asset in the (virtual) storage indexed by j when the reservation price is given. In slot t, if the ith bid is successful, DEMBID stores an additional amount of $G_{B_j}(p_i(t)) - G_{B_j}(min(p_{i-1}, \xi_i))$ into virtual storage j where the bidding price is $p_i(t)$ and the reservation price is ξ_i; otherwise, it stores nothing. Both can be stated in the following compact expression

$$\left[G_{B_j}(p_i(t)) - G_{B_j}(\min(p_{i-1}, \xi_i))\right]^+ .$$

$G_{B_j}(p)$ is decreasing in p, which gives some intuition on the storage behavior. Once the storage level is low, DEMBID is more eager to store the asset; hence, it accepts higher prices. On the other hand, at high storage utilization, it charges the storage only if the price is low.

We can illustrate the storage behavior by examining DEMBID at price endpoints. Plugging in the price $p_{\min}$, we have $G_{B_j}(p_{\min}) = B_j$, meaning that all virtual storages will be stored at their capacity B_j. Practically speaking, this indicates that the first physical inventory is fully at capacity B, while the remaining virtual storages fully satisfy their respective time varying demand $d(t)$.

In the other extreme, plugging in $\frac{p_{\max}}{\alpha}$, we have $G_{B_j}\left(\frac{p_{\max}}{\alpha}\right) = 0$, meaning that all virtual storages will aim to store nothing. Practically speaking, this means that with target storage of 0, DEMBID relies on previous inventory levels to meet demand and will not purchase beyond the required demand.

Finally, the bidding quantity q_i is determined by aggregating the value above over all virtual storages as in Lines 17–19. In the case that $p_k(t) = p(t)$, the procurement by DEMBID is equivalent to that of DEM-ON.

With the bidding quantities designed to guarantee convergence toward optimal online behavior, the bidding prices are designed around realistic worst-case performance. $p_k(t)$ is incorrect by a factor of $\frac{p_k(t)}{p(t)}$, and our analysis in Section 3.2 shows that this factor characterizes how much DEMBID degrades the competitive ratio of DEM-ON. Thus we design our bidding prices to minimize this degradation factor. DEMBID sets exponentially decreasing prices over the domain $p \in \left[p_{\min}, \frac{p_{\max}}{\alpha}\right]$, with each successive bidding price being $r_d = (\theta/\alpha)^{1/m(t)}$ times smaller than the previous bid. (Lines 12–16). In turn, this guarantees that $p_k(t)$ is no more than r_d times larger than the clearing price.

3.2. Competitive analysis of DEMBID

The following theorem gives the main result for DEMBID.

Theorem 1. *DEMBID achieves the competitive ratio of* α_{DEMBID} *as*

$$\alpha_{\text{DEMBID}} = \alpha \cdot \left(\frac{\theta}{\alpha}\right)^{1/(m-1)}$$

for $m > 1$, *where* m *is the maximum number of bids, and* α *is defined in Eq.* (12) *as the competitive ratio of DEM-ON [2].*

The competitive ratio α_{DEMBID} has several desirable qualities. First, α_{DEMBID} can interpreted as the online optimal competitive ratio α degraded by a factor of $(\theta/\alpha)^{1/(m-1)}$, which comes due to unknown price values for the incoming slot. Furthermore, this degradation factor is decreasing with the number of bids m, and α_{DEMBID} converges toward α for sufficiently large m. Lastly, by the nature of α, $\alpha_{\text{DEMBID}} \to 1$, when θ approaches 1. In what follows, we proceed to define some additional notations to facilitate the proof of Theorem 1.

Preliminary Notation. Define $\omega \in \Omega$ as an input instance including the prices and demands over the time horizon, i.e.

$$\omega = [\omega(t) = \langle p(t), d(t)\rangle]_{t \in \mathcal{T}}.$$

Consider the execution of DEMBID on input ω which produces a set of accepted bids for each $t \in \mathcal{T}$. Let k be the index of the last accepted bid at time t, which necessarily satisfies the following:

$$p_i(t) \begin{cases} \geq p(t), & \forall 1 \leq i \leq k; \\ < p(t) & \text{otherwise.} \end{cases}$$

For an input instance ω, the set of accepted bids by DEMBID are also used to define an alternative input, ω_b, as

$$\omega_b = [\omega_b(t) = \langle p_k(t), d(t)\rangle]_{t \in \mathcal{T}}.$$

Note that, associated with each input instance ω, we can always find an input ω_b based on the above construction. From DEM-ON being α-competitive, we have

$$\mathtt{cost}_\omega(\mathtt{DEM-ON}) \le \alpha \cdot \mathtt{cost}_\omega(\mathtt{OPT}) + \mathtt{cons}, \tag{13}$$

where $\mathtt{cons} \ge 0$ is a constant number, and $\mathtt{cost}_\omega(\mathtt{OPT})$ is the cost of the offline optimal algorithm on the input ω. The following definition also specifies a portion of the procurement strategy for DEM-ON.

Definition 1. For any time $t \in \mathcal{T}$, the minimum required procurement to satisfy the demand constraint is

$$x_{\min}(t) = [d(t) - b(t-1)]^+,$$

and the procurement by DEM-ON is:

$$x_{\mathtt{DEM-ON}}(t) = \max\left\{\sum_{j=1}^{v}\left[G_{B_j}(p(t)) - G_{B_j}(\xi_j)\right]^+, x_{\min}(t)\right\}. \tag{14}$$

We start to prove Theorem 1 by beginning with a result expressing the relationship between the procurement of DEMBID and DEM-ON.

Lemma 1. *For the time-specific inputs $\omega(t) = \langle p(t), d(t)\rangle$ and $\omega_b(t) = \langle p_{k(t)}(t), d(t)\rangle$,*

$$x_{\mathtt{DEMBID}}(\omega(t)) = x_{\mathtt{DEM-ON}}(\omega_b(t)).$$

Proof. Note that the deterministic bid of DEMBID also procures quantity $x_{min}(t)$. For analysis, we decompose the procurement of DEMBID into procurement from the reservation functions and procurement $x_{min}(t)$:

$$\begin{aligned} x_{\mathtt{DEMBID}}(\omega(t)) &= \sum_{i=1}^{k(t)(t)} q_i + x_{\min}(t) \\ &= \max\left\{\sum_{j=1}^{v}\left[G_{B_j}(p_{k(t)}(t)) - G_{B_j}(\xi_j)\right]^+, x_{\min}(t)\right\} \\ &= x_{\mathtt{DEM-ON}}(\omega_b(t)). \end{aligned}$$

The second equality is by construction of q_i, and the last equality is from directly substituting the input $\omega_b(t)$ into (14). □

This result will help establish the following lemma on the cost of DEMBID.

Lemma 2. *For any $\omega \in \Omega$, $\mathtt{cost}_\omega(\mathtt{DEMBID})$ is upper bounded by*

$$\mathtt{cost}_\omega(\mathtt{DEMBID}) \le \mathtt{cost}_{\omega_b}(\mathtt{DEM-ON}).$$

Proof. Consider the cost of DEMBID:

$$\begin{aligned} cost_\omega(\mathtt{DEMBID}) &= \sum_{i=1}^{T} p(t) x_{\mathtt{DEMBID}}(\omega(t)) \le \sum_{i=1}^{T} p_{k(t)}(t) x_{\mathtt{DEMBID}}(\omega(t)) \\ &= \sum_{i=1}^{T} p_{k(t)}(t) x_{\mathtt{DEM-ON}}(\omega(t)) = cost_{\omega_b}(\mathtt{DEM-ON}). \end{aligned}$$

The first inequality is due to the nature of last accepted bids $p_{k(t)}(t) \ge p(t)$. □

Lemma 3. *Let $r = \sup_{\omega\in\Omega} \max_{t\in\mathcal{T}} \frac{p_k(t)}{p(t)}$ be the maximum ratio between the last accepted bid and clearing price over the time period. Then for any $\omega \in \Omega$, DEMBID is $\alpha \cdot r$-competitive.*

Proof. We substitute (13) into the result of Lemma 2, except with input ω_b.

$$\begin{aligned} \mathtt{cost}_\omega(\mathtt{DEMBID}) &\le \mathtt{cost}_{\omega_b}(\mathtt{DEM-ON}) \\ &\le \alpha \cdot \mathtt{cost}_{\omega_b}(\mathtt{OPT}) + \mathtt{cons} \\ &\le \alpha \cdot \frac{\mathtt{cost}_{\omega_b}(\mathtt{OPT})}{\mathtt{cost}_\omega(\mathtt{OPT})} \mathtt{cost}_\omega(\mathtt{OPT}) + \mathtt{cons}. \end{aligned} \tag{15}$$

Let $x_{\texttt{OPT}}(t)$ be the optimal offline procurement under ω. Then

$$\texttt{cost}_{\omega}(\texttt{OPT}) = \sum_{t=1}^{T} p(t)x_{\texttt{OPT}}(t).$$

Since ω_b has the same demand inputs as ω and the demand constraints of CostMin only depend on demand, $x_{\texttt{OPT}}(t)$ produces a feasible solution to CostMin with input ω_b. Then, this feasible solution is an upper bound on $\texttt{cost}_{\omega_b}(\texttt{OPT})$, i.e.,

$$\texttt{cost}_{\omega_b}(\texttt{OPT}) \le \sum_{t=1}^{T} p(t)x_{\texttt{OPT}}(t) \le \sum_{t=1}^{T} p_k(t)x_{\texttt{OPT}}(t).$$

It follows from the above expressions that

$$\begin{aligned}\frac{\texttt{cost}_{\omega_b}(\texttt{OPT})}{\texttt{cost}_{\omega}(\texttt{OPT})} &\le \frac{\sum_{t=1}^{T}\frac{p_k(t)}{p(t)}p(t)x_{\texttt{OPT}}(t)}{\sum_{t=1}^{T}p(t)x_{\texttt{OPT}}(t)}\\ &\le \frac{r\sum_{t=1}^{T}p(t)x_{\texttt{OPT}}(t)}{\sum_{t=1}^{T}p(t)x_{\texttt{OPT}}(t)}\\ &= r,\end{aligned}$$

where the second inequality simply uses the definition of r. Substituting the above result to (15) completes the proof. □

As the last step, we proceed to upper bound the value of r.

Lemma 4. *When p_i is set according to the* DEMBID *algorithm, we have $r \le \left(\frac{\theta}{\alpha}\right)^{1/(m-1)}$.*

Proof. Since $p_k(t)$ is the last accepted bid, the first rejected bid is less than the clearing price, i.e. $p_{k+1}(t) < p(t)$. This helps provide an upper bound for r as follows

$$r = \sup_{\omega\in\Omega}\max_{t\in\mathcal{T}}\frac{p_k(t)}{p(t)} < \sup_{\omega\in\Omega}\max_{t\in\mathcal{T}}\frac{p_k(t)}{p_{k+1}(t)}$$

At time $t \in \mathcal{T}$, DEMBID sets $p_i(t)$ such that $p_i(t) = p_0/r_d^i$, where $r_d = (\theta/\alpha)^{1/m(t)}$. This means the ratio between two consecutive bids is fixed by $\frac{p_i(t)}{p_{i+1}(t)} = r_d$, regardless of the price and demand inputs. Thus

$$r < r_d = \left(\frac{\theta}{\alpha}\right)^{1/m(t)} \le \left(\frac{\theta}{\alpha}\right)^{1/(m-1)}$$

where the last inequality arises from $m(t)$ being either m or $m-1$ depending on the deterministic bid. □

Combining Lemmas 3 and 4 yields the result in Theorem 1.

3.3. Bidding strategy design for StrSUP

In this section, we propose SUPBID, a bidding strategy design algorithm for ProfitMax that determines how to supply the asset to the market and to charge/discharge the storage system. The simplified version of this problem when market price is known for the incoming slot is studied in [4]. Like DEMBID, we design SUPBID following an *adaptive threshold-based* strategy using virtual storage and multiple bids. Since the general flow of SUPBID is the same, in the following we highlight the differences in the algorithms.

Virtual Storage. Unlike DEMBID, SUPBID initializes the virtual storage when the physical storage is fully charged (see Line 29 in Algorithm 2). The renewal process in this setting denotes the case that there is no more free space to store the renewable storage for future use, hence, we renew the virtual storage to its initial value.

Adaptive Reservation Policy. For each (virtual) storage j with capacity B_j, an adaptive reservation policy is implemented with the following threshold function [1]:

$$F_{B_j}(p) = \frac{B_j}{\ln\theta + 1}\left(\ln\frac{p}{p_{\min}} + 1\right),\ p \in [p_{\min}, p_{max}]$$

$F_{B_j}(p)$, which is an increasing function with respect to p, refers to the target amount of the sold asset. Note that $F_{B_j}(p_{max}) = B_j$, meaning the algorithm will sell out the entire amount of asset in the storage if p_{max} is observed. We note that the above threshold function is designed such that SUP-ON achieves the best possible ratio of $\ln\theta + 1$ for the case that the price of incoming slot is known. Similarly, the usage of $\ln\theta + 1$ in the threshold function is motivated by being the same competitive ratio as the optimal online algorithm for the k-max search problem [13] as $k \to \infty$.

Algorithm 2 SUPBID: The bidding algorithm for supply-side participant, for each $t \in \mathcal{T}$.

1: // Initialization: at $t = 1$
2: $v \leftarrow 1$;
3: $B_1 \leftarrow B$;
4: $\xi_1 \leftarrow p_{\min}$;
// The main algorithm for t
5: $\mathcal{B} \leftarrow \emptyset$
6: $m(t) \leftarrow m$
// Deterministic bid: To avoid overflow of asset
7: **if** $u(t) > B - b(t-1)$ **then**
8: $\quad x_{\min} \leftarrow u(t) - B + b(t-1)$
9: $\quad \mathcal{B} \leftarrow \mathcal{B} \cup < p_{\min}, x_{\min} >$
10: $\quad m(t) \leftarrow m(t) - 1$
11: **end if**
// Decide bidding prices
12: $r_s \leftarrow \theta^{1/m(t)}$
13: $p_0 \leftarrow p_{min}$
14: **for** each $i = 1, 2, \ldots, m(t)$ **do**
15: $\quad p_i \leftarrow p_0 \cdot r_s^i$
16: **end for**
// Decide bidding quantities
17: **for** each $i = 1, 2, \ldots, m(t)$ **do**
18: $\quad q_i \leftarrow \sum_{j=1}^{v} \left[F_{B_j}(p_i) - F_{B_j}(\min(p_{i-1}, \xi_j))\right]$
19: **end for**
20: $\mathcal{B} \leftarrow \mathcal{B} \cup \left\{(p_1, q_1), (p_2, q_2), (p_3, q_3), \ldots, (p_{m(t)}, q_{m(t)})\right\}$
21: Submit $\mathcal{B}$
// Receive $p(t)$ and Update the reservation price
22: $k \leftarrow$ index of last accepted bid
23: $y(t) \leftarrow \sum_{i=1}^{k} q_i$
24: $b(t) \leftarrow b(t-1) - y(t) + u(t)$
25: **for** each $i = 1, 2, \ldots, v$ **do**
26: $\quad \xi_i \leftarrow \max\{\xi_i, p(t)\}$
27: **end for**
// Update the virtual storage
28: **if** $b(t) = B$ **then**
29: $\quad$ Initialize the algorithm by executing Lines (2)-(4)
30: **else**
31: $\quad$ **if** $u(t) > 0$ **then**
32: $\qquad v \leftarrow v + 1$
33: $\qquad B_v \leftarrow u(t)$
34: $\qquad \xi_v \leftarrow p_{\min}$
35: $\quad$ **end if**
36: **end if**

Multiple-bid strategy. If the ith bid is accepted, the jth virtual storage sells

$$\left[F_{B_j}(p(t)) - F_{B_j}(\min(p_{i-1}, \xi_j))\right]^+ .$$

Then, the total bidding quantity q_i is the aggregate amount sold over the set of available virtual storages as represented in Line 18 of Algorithm 2. The main difference in bidding prices arises from the different adaptive reservation policy. Since $F_{B_j}(p)$ is increasing, SUPBID submits exponentially increasing prices over the domain of $F_{B_j}(p)$. Each successive bidding price is $r_s = \theta^{1/m(t)}$ times larger than the previous bid. For details of generating multiple bids, we refer to Lines 7–19. In this manner, $p_k(t)$ is no more than r_s times smaller than the clearing price.

3.4. Competitive analysis for SUPBID

Theorem 2. *SUPBID achieves the competitive ratio of $\alpha_{\texttt{SUPBID}}$ as*

$$\alpha_{\texttt{SUPBID}} = (\ln\theta + 1) \cdot \theta^{1/(m-1)}$$

for $m > 1$, where m is the maximum number of bids, and $\ln\theta + 1$ is the competitive ratio of SUP-ON [1].

The structure of the proof for Theorem 2 follows the same ideas as DEMBID, so we first comment on the main result. The competitive ratio $\alpha_{\mathtt{SUPBID}}$ has several desirable properties. First, $\alpha_{\mathtt{SUPBID}}$ can interpreted as the online optimal competitive ratio $\ln\theta + 1$ degraded by a factor of $\theta^{1/(m-1)}$. Furthermore, this degradation factor is decreasing with the number of bids m, and $\alpha_{\mathtt{SUPBID}}$ converges toward $\ln\theta + 1$, i.e., the optimal competitive ratio for simplified SUP-ON [1], for sufficiently large m. Lastly, $\alpha_{\mathtt{SUPBID}} \to 1$ when θ approaches 1.

We now proceed to prove the result in Theorem 2 beginning with several lemmas.

Lemma 5. *For any $\omega \in \Omega$, $\mathtt{profit}_\omega(\mathtt{SUPBID})$ is lower bounded by*

$$\mathtt{profit}_\omega(\mathtt{SUPBID}) \geq \mathtt{profit}_{\omega_b}(\mathtt{SUP-ON}).$$

Proof. Through analysis similar to the proof of Lemma 1, we begin with the observation

$$x_{\mathtt{SUPBID}}(\omega(t)) = x_{\mathtt{SUP-ON}}(\omega_b(t)).$$

Consider the profit of SUPBID:

$$\begin{aligned}\mathtt{profit}_\omega(\mathtt{SUPBID}) &= \sum_{i=1}^{T} p(t)x_{\mathtt{SUPBID}}(\omega(t))\\ &\geq \sum_{i=1}^{T} p_{k(t)}(t)x_{\mathtt{SUPBID}}(\omega(t))\\ &= \sum_{i=1}^{T} p_{k(t)}(t)x_{\mathtt{SUP-ON}}(\omega_b(t))\\ &= \mathtt{profit}_{\omega_b}(\mathtt{SUP-ON}).\end{aligned}$$

The first inequality is due to the nature of last accepted bids $p_{k(t)}(t) \leq p(t)$, and the second equality is directly from the profit structure of SUP-ON under input ω_b. □

Lemma 6. *Let $r = \inf_{\omega\in\Omega} \min_{t\in\mathcal{T}} \frac{p_{k(t)}(t)}{p(t)}$ be the minimum ratio between the last accepted bid and clearing price over the time period and all inputs. Then for any $\omega \in \Omega$, SUPBID is $(\ln\theta + 1)\cdot r$-competitive.*

Proof. From SUP-ON being $(\ln\theta + 1)$-competitive,

$$\mathtt{profit}_\omega(\mathtt{SUP-ON}) \geq \frac{1}{\ln\theta + 1}\cdot \mathtt{profit}_\omega(\mathtt{OPT}) + c, \tag{16}$$

where $c \geq 0$ is a constant number, and $\mathtt{profit}_\omega(\mathtt{OPT})$ is the profit of the offline optimal algorithm on the input ω. We substitute (16) into the result of Lemma 5, except with input ω_b.

$$\begin{aligned}\mathtt{profit}_\omega(\mathtt{SUPBID}) &\geq \mathtt{profit}_{\omega_b}(\mathtt{SUP-ON})\\ &\geq \frac{1}{\ln\theta + 1}\cdot \mathtt{profit}_{\omega_b}(\mathtt{OPT}) + c\\ &\geq \frac{1}{\ln\theta + 1}\cdot \frac{\mathtt{profit}_{\omega_b}(\mathtt{OPT})}{\mathtt{profit}_{\omega}(\mathtt{OPT})}\mathtt{profit}_\omega(\mathtt{OPT}) + c.\end{aligned} \tag{17}$$

For clarity, let $x_{\mathtt{OPT}}(t)$ be the optimal offline amount sold under ω. Then

$$\mathtt{profit}_\omega(\mathtt{OPT}) = \sum_{t=1}^{T} p(t)x_{\mathtt{OPT}}(t).$$

Because ω_b has the same demand inputs as ω and the demand constraints of ProfitMax only depend on demand, $x_{\mathtt{OPT}}(t)$ produces a feasible solution to ProfitMax with input ω_b. Then this feasible solution is a lower bound on $\mathtt{profit}_{\omega_b}(\mathtt{OPT})$:

$$\mathtt{profit}_{\omega_b}(\mathtt{OPT}) \geq \sum_{t=1}^{T} p_{k(t)}(t)x_{\mathtt{OPT}}(t).$$

It follows from the above equation that

$$\begin{aligned}\frac{\texttt{profit}_{\omega_b}(\texttt{OPT})}{\texttt{profit}_{\omega}(\texttt{OPT})} &\geq \frac{\sum_{t=1}^{T} p_{k(t)}(t)x_{\texttt{OPT}}(t)}{\sum_{t=1}^{T} p(t)x_{\texttt{OPT}}(t)} \\ &= \frac{\sum_{t=1}^{T} \frac{p_{k(t)}(t)}{p(t)} p(t)x_{\texttt{OPT}}(t)}{\sum_{t=1}^{T} p(t)x_{\texttt{OPT}}(t)} \\ &\geq \frac{r\sum_{t=1}^{T} p(t)x_{\texttt{OPT}}(t)}{\sum_{t=1}^{T} p(t)x_{\texttt{OPT}}(t)} \\ &= r,\end{aligned}$$

where the second inequality simply uses the definition of r. Substituting the above equation to (17) completes the proof. □

Lemma 7. *When p_i is set according to the SUPBID algorithm, we have $r \geq \theta^{1/(m-1)}$.*

Proof. Since $p_{k(t)}(t)$ is the last accepted bid, the first rejected bid is less than the clearing price, i.e. $p_{k(t)+1}(t) < p(t)$. This helps provide a lower bound for r as follows

$$r = \min_{t\in\mathcal{T}} \frac{p_{k(t)}(t)}{p(t)} > \min_{t\in\mathcal{T}} \frac{p_{k(t)}(t)}{p_{k(t)+1}(t)}.$$

At time $t \in \mathcal{T}$, SUPBID sets $p_i(t)$ as:

$$r_s = \theta^{1/m(t)}$$

$$p_i(t) = p_0 \cdot r_s^i.$$

which means the ratio between two consecutive bids is fixed by $\frac{p_i(t)}{p_{i+1}(t)} = r_s$. Thus

$$r > r_s = \theta^{1/m(t)} \geq \theta^{1/(m-1)}$$

where the last inequality is from $m(t)$ being either m or $m-1$, depending on if the deterministic bid is submitted. □

Combining Lemmas 6 and 7 yields the result in Theorem 2.

4. Empirical evaluation

In this section, we evaluate the performance of our algorithms DEMBID and SUPBID using real-world data traces. Additional overview and background on the electricity market are included in the Appendix. Our experiments will answer the following questions:

(1) How do DEMBID and SUPBID compare to basic algorithms that know the market price for the incoming slot? Our result shows that the online strategies DEMBID and SUPBID are only 5% (on average) worse than the algorithms that know the price in advance, and this performance gap shrinks as the number of bids grows.

(2) How do DEMBID and SUPBID compare to the bidding algorithms that do not utilize virtual storage in decision making? Our result shows that by adding the virtual storage in decision making the performance will increase by at least 10%, on average.

(3) How sensitive are DEMBID and SUPBID with respect to the parameters as the penetration of renewable and capacity? Our results demonstrate that DEMBID and SUPBID outperform the alternative algorithms in a wide range of parameter settings.

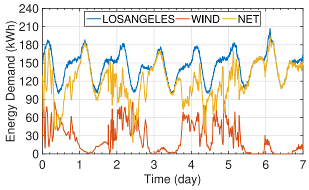
4.1. Experimental methodology

Settings. We use energy demand traces from Akamai clusters [14], renewable data generations (wind and solar) [8,9], and electricity prices from NYISO [6]. We set the length of time slot to 5 min based on the recent FERC rule [10]. The time horizon is 1 day ($T = 12 * 24 = 288$). Our data traces for renewable have the resolution of 15 min or 1 h. Hence, in our experiments, we down-sample the renewable data to 5-minute slots. We set the capacity of energy storage to $C = 18 * \max_{t\in T} d(t)$ (resp. $C = 18 * \max_{t\in T} u(t)$), adequate storage to cover the demand (resp. to store renewable output) for at least 1.5 h. The default renewable penetration for wind and solar is 50%. Finally, each data point in figures corresponds to the average results of 365 days over a year.

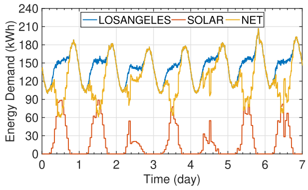
▷ *Energy Demand.* For CostMin we need energy demand of a large energy customer. Toward this, we use energy demand traces from Akamai's server clusters [7]. The data includes the server load information of several worldwide data centers during one month collected every 5 min, and we use the information of an Akamai data center in New York city.

Table 2
Summary of algorithms that are evaluated.

Our proposed online algorithms	
DEMBID	The online bidding strategy for StrDEM (Algorithm 1)
SUPBID	The online bidding strategy for StrSUP (Algorithm 2)
sDEMBID	A simplified version of DEMBID without virtual storage
sSUPBID	A simplified version of SUPBID without virtual storage
Other algorithms for comparison	
OPT	Optimal offline solution with storage for CostMin and ProfitMax
DEM-ON [2]	The existing online algorithm for CostMin with known $p(t)$ (ARP in the original paper)
SUP-ON [4]	The existing online algorithm for ProfitMax with known $p(t)$ (sOffer in the original paper)



(a) Net energy demand with wind



(b) Net energy demand with solar

Fig. 2. The net energy demand with on-site renewables.

To calculate energy consumption as a function of load, we use the standard linear model in [15,16]. We note that while we use data center energy demand, the proposed algorithms are general and could be used for other loads.

▷ *Renewable Data Traces.* We gather the renewable data traces for both wind and solar. The wind data traces are from Eastern and Western Data Sets with 100 MW to 1,435 MW Output Point Capacity [9]. The hourly solar data is from PVWatts [8]. We picked the data from stations that are less than 80 miles from the New York City, where we have our energy demands.

▷ *Real-time Energy Prices:* Finally, we use the real-time energy prices from NYISO [6], and a sample representation of the time-varying energy prices is given in Fig. 1.

Performance Metric. For CostMin, we report the cost ratio between the cost of different algorithms divided by the cost of the offline optimal solution denoted by OPT. Similarly, for ProfitMax, we report the profit ratio between the profit of offline optimal divided by the cost of different algorithms. For both CostMin and ProfitMax versions, the ratios are greater than 1 and the lower the ratio, the better the performance.

Comparison Algorithms. We compare the proposed bidding algorithms with several algorithms as listed in Table 2. First, we note that by using cost and profit ratios as the performance metric, we implicitly compare our algorithms with the offline optimum (OPT). Furthermore, we compare with two categories of algorithms:

▷ sDEMBID and sSUPBID: Both DEMBID and SUPBID algorithms are designed based on the virtual storage for improved performance. To show the impact of virtual storage, we also implemented a simplified version of both algorithms, called sDEMBID and sSUPBID, in which we simply consider utilizing just one physical storage.

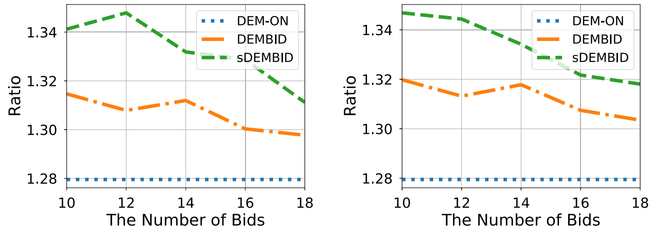
▷ DEM-ON (ARP in [2]) and SUP-ON (sOffer in [4]): Both algorithms are proposed in the previous work, and both assume that the market price is known before the current slot. Since both algorithms have more information for decision making, they are expected to achieve better performance than SUPBID and DEMBID. The gap in performance is due to additional cost of not knowing the current price, and we investigate how to close this gap by submitting more bids.

4.2. Evaluation results for DEMBID and SUPBID

We evaluate the performance of DEMBID (SUPBID) by varying the number of bids in Fig. 3 (Fig. 4) from 10 to 18 and varying the capacity of storage in Fig. 5 (Fig. 6). For DEMBID, we further change the penetration rate of renewables and report the results in Fig. 7. Finally, we investigate the impact of rate constraints for SUPBID in Fig. 8.

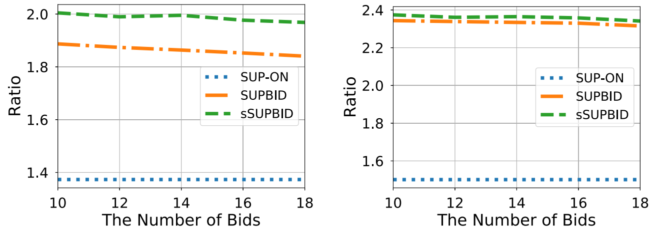
▷ *Wind vs. Solar.* The results in Figs. 3, 5, and 7 for DEMBID show that for the cases with wind as the local renewable, the cost ratios are slightly worse than the solar in most cases. This is because the wind output is much more unpredictable than the solar output, as demonstrated in Fig. 2. This increased uncertainty leads to worse cost ratios for wind. The same observation is visible for the profit ratios of SUPBID in Figs. 4, 6, and 8.

▷ *Comparison of DEMBID and SUPBID vs. sDEMBID and sSUPBID.* In most experiments, both SUPBID and DEMBID substantially outperform sSUPBID and sDEMBID with only one storage. For example, in Fig. 3, the average cost ratios of DEMBID and sDEMBID are 1.31 and 1.34 at 10 bids, demonstrating that DEMBID outperforms by 2.3%.



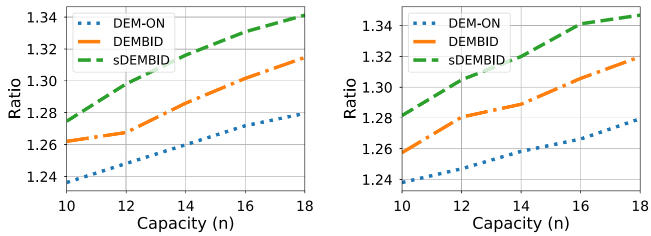
(a) Wind (b) Solar

Fig. 3. The impact of increasing bids on DEMBID.



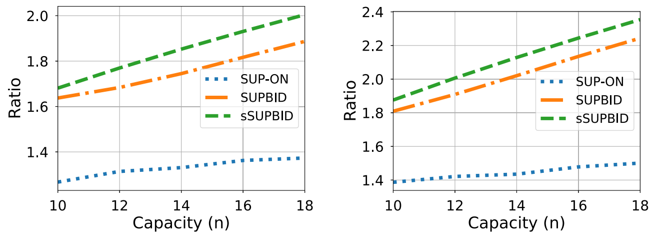
(a) Wind (b) Solar

Fig. 4. The impact of increasing bids on SUPBID.



(a) Wind (b) Solar

Fig. 5. The impact of capacity on DEMBID.



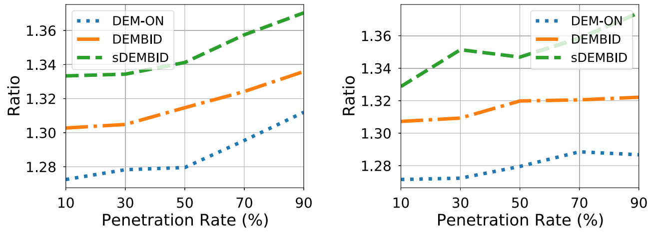
(a) Wind (b) Solar

Fig. 6. The impact of capacity on SUPBID.

▷ *Comparison of DEMBID and SUPBID vs. DEM-ON and SUP-ON.* Fig. 3 (Fig. 4) show that at 10 bids, the average cost (profit) ratio of DEMBID (SUPBID) is only 2.3% (32.1%) worse than DEM-ON (SUP-ON). The performance degradation of both algorithms decreases as the number of bids increases.

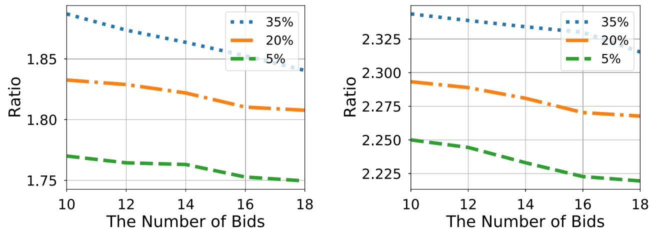
▷ *Impact of Number of Bids.* The results in Figs. 3 and 4 demonstrate that when the number of bids increases, the performance of DEMBID and SUPBID improve. By ranging from 10 to 18 bids, DEMBID drops from 1.315 to 1.298 and SUPBID drops from 2.4 to 2.2. This improved performance is aligned with the theoretical guarantees of SUPBID and DEMBID improving with the number of bids.

▷ *Impact of Storage Capacity.* In these experiments, we fix the number of bids to 10 and vary the storage capacity. We define $n = C/\max_{t\in\mathcal{T}} d(t)$ (resp. $n = C/\max_{t\in\mathcal{T}} u(t)$) as the ratio between the capacity of storage and the maximum



(a) Wind (b) Solar

Fig. 7. The impact of renewable penetration on `DEMBID`.



(a) Wind (b) Solar

Fig. 8. The impact of rate constraints on `SUPBID`.

demand (resp. maximum output). The results in Figs. 5 and 6 demonstrate that the cost and profit ratios of all online algorithms increase as the capacity increases. This is reasonable since with increased capacity, `OPT` has more design space for cost minimization or profit maximization, making it harder for online algorithms to compete.

▷ *Impact of Renewable Penetration.* In CostMin we vary the penetration rate from 10% to 90%. The results in Fig. 7 demonstrate that with increased renewable penetration, the cost ratios of `DEMBID` and `sDEMBID` increase due to the increased uncertainty of renewables.

▷ *Impact of Rate Constraints.* Lastly, we investigate the impact of different values of rate constraints on `SUPBID`. For simplicity, we consider identical charging and discharge rates, and normalize them as a percentage of the storage capacity as a common approach in characterizing battery specs. [17]. We choose three common categories of storages in data centers based on their rate to capacity ratios [18,19]: (1) Compressed Air Energy Storage (CAES) with ratio of 5%; (2) Lead-Acid (LA) with 20%; (3) Lithium-Ion (LI) with 35%. In Fig. 8 we report the profit ratios of `SUPBID` for different values of rate to capacity ratio while also varying the number of bids. The results show that for higher rate to capacity ratio, the profit ratio is also higher. This is due to a similar effect observed in the capacity experiments — `OPT` is less constrained with higher rate to capacity ratios and has more design space, so `DEMBID` performs relatively worse.

5. Concluding remarks

In this paper, we developed bidding strategies for online linear optimization with inventory management constraints with the application of storage-assisted participants in the supply and demand sides of the electricity market. Our algorithms are built on top of the state-of-the-art algorithms for managing inventories in dynamic environments in terms of market pricing and supply and demand. We tackled unique challenges of unknown price, possibility of failure of bids,and uncertain asset trading quantity from multiple bid submissions. We analyzed the competitive ratios of our online bidding strategies and showed that they converge to the best possible values with sufficiently large amounts of bids. Our extensive experimental evaluations using real-world data traces showed that the performance of our algorithms is within 5% of the algorithms that know the market price in advance, and outperform alternative algorithms by more than 10%. As future work, we plan to extend our empirical study using the data from other locations to see the impact of different markets on the performance of the algorithms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support from the U.S. National Science Foundation (NSF) under grant numbers CNS-1908298, CNS-1763617, NGSDI-2105494, CPS-2136199, and CAREER-2045641.

Appendix. An overview of electricity market

In this section, we provide the necessary background of deregulated electricity markets related to our work.

The supply chain. The supply chain of a deregulated electricity market is illustrated in Fig. A.9 where there are three main components:

▷ In the middle, as the trader, there is an *independent system operator* (a.k.a., ISO, e.g., NYISO, CAISO, PJM, etc.), that provides the trading place. ISO matches supply bids from suppliers and demand bids from utilities and customer by running auction mechanisms.

▷ Customers, that are typically *utility* companies or in some cases large energy customers, who submit bids to buy electricity from the market in order to serve a group of retail customers or its large energy demand.

▷ Suppliers or *generation companies* such as power plants who submit supply bids to sell electricity to market.

Market operations. Markets usually operate in multi-settlement manner. That is, the operations are done in different settlement periods. The common approach is to have two-settlement market that includes day-ahead and real-time operations. In the day-ahead market operation, suppliers and utilities submit their bids for the trades in the next day on an hourly basis. On the other hand, the real-time electricity market usually operates in 5-minutes to 1 h basis. In this paper, we focus on real-time operations of the market.

Bid submission. The bidding submission in real-time operation is as follows. Shortly before the operation time, the suppliers and customers submit their bids, including the bidding price and the bidding quantity, for the forthcoming slot. After receiving the bids from supply and demand parties, ISO matches the bids from both sides, runs a double auction mechanism [20], and announces a market clearing price. Then, for a supply bid, if the bidding price is less than or equal to the clearing price, its quantity is considered as the commitment to the market for the next slot. The supplier is paid according to the clearing price. Similarly, for the demand side, if the bidding price is greater than or equal to the clearing price, the bid is accepted and the bidder buys its bidding quantity from the market. Given the above scenario, *the goal of bidding strategies is to determine the bidding price and quantity such that the long-term profit of the supplier is maximized, or equivalently, the long-term cost of the customer is minimized.*

Storage-assisted participants. We consider storage-assisted participants for both sides of the market, as depicted in Fig. A.9. On the supply side, we consider a supplier with renewable resources with on-site storage systems. We consider this participant as a single entity with renewable and storage, and the bidding strategy determines how to sell the renewable output to the market and/or charge/discharge the storage. Similarly, on the demand side, we consider an energy customer like a data center that is large enough to participate in the market. While in experiments, we use data center energy demands from data centers, our model is general and could be applied to any other type of energy customers or utilities that participate in the market. We again consider that the energy of the large customer could be satisfied from multiple resources such as local renewables, and on-site energy storage systems. Then, the bidding strategy determines how to manage the storage and cover the energy by buying from the market.

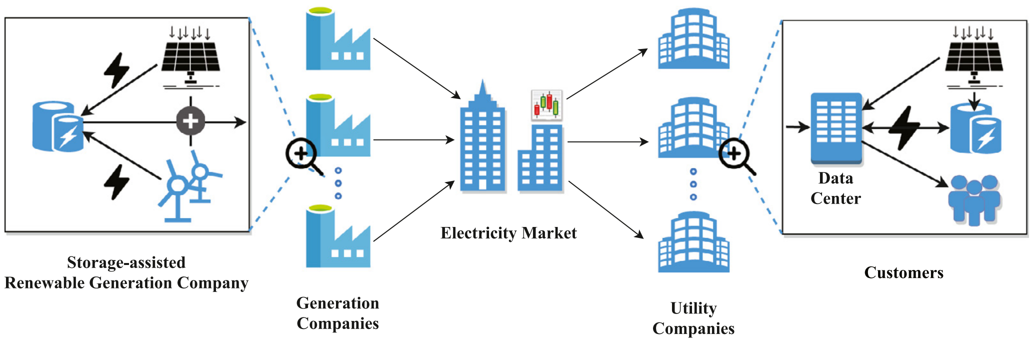


Fig. A.9. The supply chain of a deregulated electricity market with storage-assisted participants.

References

[1] L. Yang, W.S. Wong, M.H. Hajiesmaili, An optimal randomized online algorithm for QoS buffer management, Proc. ACM Meas. Anal. Comput. Syst. 1 (2) (2017) 36:1–36:26.

[2] L. Yang, M. Hajiesmaili, R. Sitaraman, A. Wierman, E. Mallada, W.S. Wong, Online linear optimization with inventory management constraints, ACM SIGMETRICS Perform. Eval. Rev. 48 (2020) 7–8, http://dx.doi.org/10.1145/3410048.3410053.

[3] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.

[4] L. Yang, M.H. Hajiesmaili, H. Yi, M. Chen, Hour-ahead offering strategies in electricity market for power producers with storage and intermittent supply, in: Proc of. ACM SIGMETRICS, 2017, pp. 21–22.

[5] PJM electricity market, 2021, available at http://www.pjm.com, (Accessed May 2021).

[6] NYISO electricity market, 2021, available at http://www.nyiso.com, (Accessed May 2021).

[7] E. Nygren, R.K. Sitaraman, J. Sun, The Akamai network: a platform for high-performance Internet applications, Oper. Syst. Rev. 44 (3) (2010) 2–19.

[8] A. Dobos, PVWatts Version 5 Manual, National Renewable Energy Laboratory Golden, CO, 2014.

[9] Eastern and western data sets, 2006, available at https://www.nrel.gov/grid/eastern-western-wind-data.html, (Accessed May 2021).

[10] U.H. E. Ela, Wholesale electricity market design initiatives in the United States: Survey and research needs, in: EPRI, Technical Results, 2016, available at https://Www.Epri.Com/Pages/Product/3002009273/.

[11] E.D. Castronuovo, J. Lopes, On the optimization of the daily operation of a wind-hydro power plant, IEEE Trans. Power Syst. 19 (3) (2004) 1599–1606.

[12] J. González, D. Muela, R. Ruiz, L. Santos, A. González, Stochastic joint optimization of wind generation and pumped-storage units in an electricity market, IEEE Trans. Power Syst. 23 (2) (2008).

[13] J. Lorenz, K. Panagiotou, A. Steger, Optimal algorithms for k-search with application in option pricing, Algorithmica 55 (2008) 311–328.

[14] Akamai, 2021, available at https://www.akamai.com, (Accessed May 2021).

[15] L.A. Barroso, U. Hölzle, The case for energy-proportional computing, Computer 40 (12) (2007) 33–37, http://dx.doi.org/10.1109/MC.2007.443.

[16] V. Gupta, P. Shenoy, R.K. Sitaraman, Combining renewable solar and open air cooling for greening internet-scale distributed networks, in: Proceedings of the Tenth ACM International Conference on Future Energy Systems, ACM EEnergy, 2019, pp. 303–314.

[17] M. Team, et al., A Guide to Understanding Battery Specifications, Tech. Rep., MIT, 2008.

[18] S. Chalise, A. Golshani, S.R. Awasthi, S. Ma, B.R. Shrestha, L. Bajracharya, W. Sun, R. Tonkoski, Data center energy systems: Current technology and future direction, in: 2015 IEEE Power Energy Society General Meeting, 2015, pp. 1–5, http://dx.doi.org/10.1109/PESGM.2015.7286420.

[19] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, H. Fathy, Energy storage in datacenters: What, where, and how much? in: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 187–198, http://dx.doi.org/10.1145/2254756.2254780, URL https://doi.org/10.1145/2254756.2254780.

[20] M. Liu, F.F. Wu, Risk management in a competitive electricity market, Int. J. Electr. Power Energy Syst. 29 (9) (2007) 690–697.