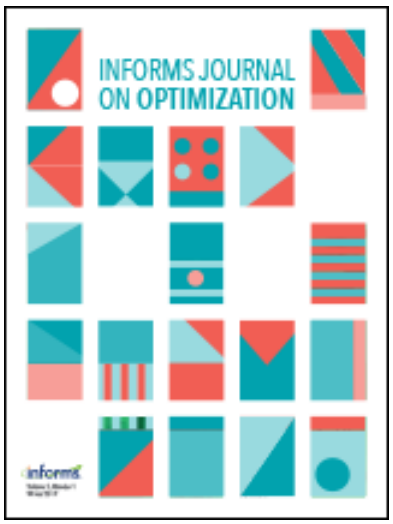


INFORMS Journal on Optimization

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Function Design for Improved Competitive Ratio in Online Resource Allocation with Procurement Costs

Mitas Ray, Omid Sadeghi, Lillian J. Ratliff, Maryam Fazel

To cite this article:
Mitas Ray, Omid Sadeghi, Lillian J. Ratliff, Maryam Fazel (2024) Function Design for Improved Competitive Ratio in Online Resource Allocation with Procurement Costs. INFORMS Journal on Optimization

Published online in Articles in Advance 23 Dec 2024

. https://doi.org/10.1287/ijoo.2021.0012

https://pubsonline.informs.org/journal/ijoo

INFORMS JOURNAL ON OPTIMIZATION
Articles in Advance, pp. 1–18
ISSN 2575-1484 (print), ISSN 2575-1492 (online)

Function Design for Improved Competitive Ratio in Online Resource Allocation with Procurement Costs

Mitas Ray,[a,*] Omid Sadeghi,[a] Lillian J. Ratliff,[a] Maryam Fazel[a]

[a] Department of Electrical & Computer Engineering, University of Washington, Seattle, Washington 98195
*Corresponding author
Contact: mitas.ray@gmail.com, https://orcid.org/0000-0001-6100-0096 (MR); omids@uw.edu (OS); ratliffl@uw.edu (LJR); mfazel@uw.edu (MF)

Received: May 12, 2021
Revised: June 6, 2022
Accepted: May 21, 2024
Published Online in Articles in Advance: December 23, 2024

https://doi.org/10.1287/ijoo.2021.0012

Abstract. We study the problem of online resource allocation, where customers arrive sequentially, and the seller must irrevocably allocate resources to each incoming customer while also facing a prespecified procurement cost function over the total allocation. The objective is to maximize the reward obtained from fulfilling the customers' requests sans the cumulative procurement cost. We analyze the competitive ratio of a primal-dual algorithm in this setting and develop an optimization framework for designing a surrogate function for the procurement cost to be used by the algorithm to improve the competitive ratio of the primal-dual algorithm. We use the optimal surrogate function for polynomial procurement cost functions to improve on previous bounds. For general procurement cost functions, our design method uses quasiconvex optimization to find optimal design parameters. We then implement the design techniques and show the improved performance of the algorithm in numerical examples. Finally, we extend the analysis by devising a posted pricing mechanism in which the algorithm does not require the customers' preferences to be revealed.

Funding: M. Fazel's work was supported in part by the National Science Foundation [Awards 2023166, 2007036, and 1740551].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/ijoo.2021.0012.

Keywords: competitive ratio • primal-dual • online optimization

1. Introduction

In the online resource allocation problem, a seller allocates D types of resources to T incoming customers. The tth customer has a payment function, denoted $v_t : \mathbb{R}_+^D \to \mathbb{R}_+$, which satisfies a natural set of assumptions listed in Assumption 1 (differentiable, concave, and monotonically increasing; see a more detailed statement in Section 3). The payment function reveals how much a customer will pay for any assigned bundle of resources. The seller has a procurement cost function, denoted by $f : \mathbb{R}_+^D \to \mathbb{R}_+$, which represents the cost incurred by the seller in procuring the resources in the cumulative allocation and is known to the seller a priori. The procurement cost function satisfies Assumption 2 (differentiable, convex, and monotonically increasing; see a more detailed statement in Section 3). We use $\mathbf{x}_t \in [0,1]^D$ to denote the bundle allocated to customer t, where the dth entry of $\mathbf{x}_t$ represents the amount of the dth type of resource in this bundle. The goal of the seller is to maximize the revenue collected from the assigned bundles to the customers minus the procurement cost of the cumulative allocation. Had the seller known the T customers' payment functions beforehand, then the optimal allocation would be the result of the following offline optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{t=1}^{T} v_t(\mathbf{x}_t) - f\left(\sum_{t=1}^{T} \mathbf{x}_t\right) \\ \text{subject to} \quad & \mathbf{0} \preceq \mathbf{x}_t \preceq \mathbf{1} \quad \forall\, t \in [T], \end{aligned} \tag{P-1}$$

where $\mathbf{x}_t \in [0,1]^D$ for $t = 1, \ldots, T$ is the optimization variable. The challenge of online resource allocation comes from its *online* nature; the seller does not have any knowledge of the future customers and must make an irrevocable allocation upon the arrival of each customer.

Online resource allocation has been studied extensively in the setting of fixed resource capacities such as Blumrosen and Nisan (2007), Chakraborty et al. (2013), and Tan et al. (2020), where there is a hard budget for each type of resource, and the unlimited supply setting as in the works of Balcan et al. (2005), where the seller has unlimited access to each resource type. In the work of Balcan et al. (2008), the authors consider both the fixed

resource capacity and unlimited capacity setting. However, in many real-world situations, additional resources may be procured albeit at increasing marginal costs, such as energy costs for running computer processors as described in Makarychev and Sviridenko (2014) and Andrews et al. (2016) and hiring costs for skilled labor as explained in Blatter et al. (2012). This motivates the problem of online resource allocation with procurement costs introduced by Blum et al. (2011).

Past literature, such as Blum et al. (2011) and Huang and Kim (2018), consider the procurement cost function to be *separable*—that is, the total cost incurred is the sum of the individual procurement costs for each resource. The work of Blum et al. (2011), which was further improved in Huang and Kim (2018), propose an online mechanism in which the seller determines the price of a particular item as a function of how much has already been sold, and the customer then chooses the bundle that maximizes their valuation function. In both works, it is assumed that the procurement cost function is separable, so the cost of procuring one item has no effect on the cost of procuring another. However, in a real-world setting, there may exist limited procurement infrastructure where procuring one resource does affect the cost of procuring another. It is thus important to generalize this setting and consider procurement cost functions that are *nonseparable*.

Although the work of Chan et al. (2015) studies the setting of nonseparable procurement costs, the assumptions made essentially restrict the procurement cost function to polynomials. Therefore, the class of more general (nonpolynomial) separable cost functions has not been addressed, and there is no strategy on how to handle procurement cost functions that do not meet the stringent assumptions in Chan et al. (2015). We, on the other hand, in Theorem 3, drop the assumptions that restrict the function class to polynomials allowing us to consider the general nonseparable case. In Online Appendix E, we provide a concrete example that highlights the polynomial restriction in Chan et al. (2015).

Many algorithms in this setting are primal-dual algorithms, which come from updating the dual variable at each time step and using it to assign the primal variable as seen throughout the literature such as in Buchbinder et al. (2007), Buchbinder and Naor (2009), Devanur and Jain (2012), Agrawal et al. (2014), Azar et al. (2016), and Eghbali and Fazel (2016). A key measure of algorithm performance in online optimization is the competitive ratio, which is defined as the ratio of the objective value achieved by the algorithm to the offline optimum (see Section 3.1). The competitive ratio we consider is under the adversarial arrival order, where the seller does not know the arriving customers or the order of their arrival. For more details on different arrival models, we refer readers to section 2.2 in Mehta (2013).

The problem of online resource allocation appears often in the operations research community for problems like airline revenue management, as described in Hwang et al. (2021) and Jaillet and Lu (2012), hospital appointment scheduling, as in Legrain and Jaillet (2013) and Erdogan et al. (2015), and bidding in auctions, as in Bertsimas et al. (2009), among others. However, many of the underlying assumptions in these problems are different from the ones we make in our setting. For example, Hwang et al. (2021) considers the arrival time of a fraction of agents to be chosen by an adversary, whereas the remaining agents come at random times. The optimization problems are also formulated differently for each setting; for example, Legrain and Jaillet (2013) consider a linear objective with budget constraints. Although many of these differences seem minor, the overall problem changes enough to not be directly captured by our formulation. Nonetheless, these setups encourage us to scrutinize our assumptions to capture many problem settings. Section 1.3 enumerates a few motivating applications of the framework proposed in this paper. For more details on related work, see Section 8.

1.1. Contributions

We analyze a greedy primal-dual algorithm, formalized in Algorithm 1 in which a surrogate function is used in place of the procurement cost function to optimize the performance of the algorithm. We discuss a simple example in Section 6 to show that the competitive ratio of the greedy primal-dual algorithm without a surrogate function approaches zero asymptotically, illustrating the need for a surrogate function. Our main contributions come in the design of the surrogate function.

- For polynomial procurement cost functions, we design a surrogate function to be used in the algorithm that achieves a better competitive ratio than algorithms proposed in existing literature (Chan et al. 2015), in particular, our competitive ratio has better dependence on the degree of the polynomial. A thorough comparison with prior work is presented at the end of Section 5.1.
- For general procurement cost functions, we write the surrogate function design problem as a quasiconvex optimization problem in which the optimization variables define the surrogate function. This strategy comes from adopting an optimization perspective for maximizing the competitive ratio similar to Eghbali and Fazel (2016). This technique allows us to construct surrogate functions for a wide class of procurement cost functions beyond

those that are separable, as in Huang and Kim (2018), and polynomials, as in Chan et al. (2015). Our result is stated formally in Theorem 3.

- Because Algorithm 1 solves a saddle-point problem at every round, it may not be practical in many situations. We therefore propose Algorithm 2 that updates the primal and dual variables sequentially. This algorithm can be interpreted as a posted pricing mechanism and is therefore incentive compatible. We extend the quasiconvex surrogate function design technique to this algorithm. Our results are stated formally in Theorems 5 and 7.

We complement our theoretical results with simulations in which we implement our design techniques on a numerical example and show better performance over prior work.

1.2. Organization

This paper is organized as follows. We close this section with a few motivating examples to show the generality of our framework. Section 2 covers the preliminaries, and the formal problem statement and primal-dual algorithm are introduced in Section 3. We analyze the competitive ratio for our primal-dual algorithm in Section 4 and then propose our surrogate function design techniques in Section 5. In Section 6, we implement our design techniques on a numerical example. In Section 7, we extend the competitive analysis and design techniques to another primal-dual algorithm that computes the primal and dual variables sequentially. A comprehensive overview of related work in the literature is provided in Section 8.

1.3. Motivating Examples

To illustrate applicability, we provide several online resource allocation problems that can be cast in the proposed framework described in Problem (P-1). In each application, we describe the incoming valuation functions $\mathbf{v}_t$, the cost function f, and what our decision vector at time t—that is, $\mathbf{x}_t$, represents.

1.3.1. Online Auction. A seller has a set of D items and T customers arrive sequentially. Let $\mathbf{x}_t \in [0,1]^D$ represent the decision vector at time t representing the bundle allocated to customer t. Each item can be included in a bundle at most once. Hence, the decision vector is constrained to $\mathbf{0} \preceq \mathbf{x}_t \preceq \mathbf{1}$. The payment function $v_t : \mathbb{R}_+^D \to \mathbb{R}_+$ is revealed by the tth customer upon arrival. The procurement function is denoted $f : \mathbb{R}_+^D \to \mathbb{R}_+$. The objective of the seller is to maximize their profit—that is, the sum of the payments of the customers minus the procurement cost of the total allocation. Variations of this framework are discussed in Bartal et al. (2003), Chan et al. (2015), and Huang and Kim (2018).

1.3.2. Data Market. A manager supervises a set of D experts with differing expertise. Data analysis tasks, such as classifying medical images, arrive online sequentially and each task can be assigned to any subset of the experts. Upon arrival, task t reveals a vector $\mathbf{c}_t$ where $[\mathbf{c}_t]_d$ quantifies the value that expert d would provide the manager if assigned to task t. The value function is linear—that is, $v_t(\mathbf{x}_t) = \mathbf{c}_t^\top \mathbf{x}_t$. When a task is assigned to an expert, the amount of time they are being paid to spend on it is bounded. Therefore, the decision vector is constrained to $\mathbf{0} \preceq \mathbf{x}_t \preceq \mathbf{1}$. The manager is responsible for paying for the experts' time and the resources needed for the experts to do their work, which is captured in a cost function $f : \mathbb{R}_+^D \to \mathbb{R}_+$. The cost of hiring skilled labor is marginally increasing and follows a convex cost function, as described in Blatter et al. (2012). The objective of the manager is to maximize the value of the completed work minus the cost of getting the work completed. Variations of this application are mentioned in Ho and Vaughan (2012).

1.3.3. Network Routing with Congestion. A network routing agent has a set of D pairs of terminals and T users arrive online with valuation functions over these routed connections. Because each pair of terminals can be assigned to a user at most once, the decision vector $\mathbf{x}_t$ is constrained to $\mathbf{0} \preceq \mathbf{x}_t \preceq \mathbf{1}$. Let $v_t : \mathbb{R}_+^D \to \mathbb{R}_+$ represent the payment function that each customer reveals upon arrival and let $f : \mathbb{R}_+^D \to \mathbb{R}_+$ denote the congestion cost function that can represent the energy needed to maintain the routed connections. Because energy usage follows diseconomies of scale—that is, energy usage is superlinear in terms of processor speed as described in Makarychev and Sviridenko (2014) and Andrews et al. (2016), f satisfies Assumption 2. The objective of the network routing agent is to maximize the valuations of the customers minus the energy costs of the cumulative assignment. Variations of this framework are discussed in Blum et al. (2011).

2. Preliminaries

In this section, we review mathematical preliminaries as needed for the technical results. Throughout, we will use boldface symbols to denote vectors. For a D-dimensional vector $\mathbf{u} \in \mathbb{R}^D$, let u_i, or equivalently $[\mathbf{u}]_i$, denote the

ith entry. The inner product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ is denoted $\langle \mathbf{u}, \mathbf{v} \rangle$ or, equivalently, $\mathbf{u}^\top \mathbf{v}$. The generalized inequality with respect to the nonnegative orthant is denoted $\mathbf{u} \geq \mathbf{v}$, and is equivalent to $u_i \geq v_i$ for all i. Define $\mathbf{1}[\mathbf{u} \geq \mathbf{v}]$ to be the vector where the ith entry equals one if $u_i \geq v_i$ and zero otherwise. The index set $\{1, \ldots, K\}$ is denoted $[K]$. Several function properties are needed for the analysis in this paper. A function $f : \mathbb{R}^D \to \mathbb{R}$ is separable if it can be written as $f(\mathbf{u}) = \sum_{i=1}^D f_i(u_i)$. A function $f : \mathbb{R}^D \to \mathbb{R}$ is quasiconvex if $\text{dom}(f)$ is a convex set and for each $\alpha \in \mathbb{R}$, the sublevel set, $S_\alpha = \{\mathbf{u} \in \text{dom}(f) | f(\mathbf{u}) \leq \alpha\}$ is a convex set.

Given a function $f : \mathbb{R}^D \to \mathbb{R}$, its convex conjugate $f^* : \mathbb{R}^D \to \mathbb{R}$ is defined be

$$f^*(\mathbf{v}) = \sup_{\mathbf{u}} \ \mathbf{v}^\top \mathbf{u} - f(\mathbf{u}).$$

For any function f and its convex conjugate f^*, the Fenchel-Young inequality holds for every $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$:

$$f(\mathbf{u}) + f^*(\mathbf{v}) \geq \mathbf{u}^\top \mathbf{v}. \tag{1}$$

For a differentiable, closed and convex function f, its gradient is given by $\nabla f(\mathbf{u}) = \arg\max_{\mathbf{v}} \mathbf{v}^\top \mathbf{u} - f^*(\mathbf{v})$, and furthermore, $f^{**} = f$. Letting $\mathbf{v} = \nabla f(\mathbf{u})$, the Fenchel-Young inequality holds with equality:

$$f(\mathbf{u}) + f^*\Big(\nabla f(\mathbf{u})\Big) = \mathbf{u}^\top \nabla f(\mathbf{u}). \tag{2}$$

Similarly, given a function $g : \mathbb{R}^D \to \mathbb{R}$, the concave conjugate $g_* : \mathbb{R}^D \to \mathbb{R}$ is defined by

$$g_*(\mathbf{v}) = \inf_{\mathbf{u}} \ \mathbf{v}^\top \mathbf{u} - g(\mathbf{u}).$$

An analogous inequality to (1) holds: $g(\mathbf{u}) + g_*(\mathbf{v}) \leq \mathbf{u}^\top \mathbf{v}$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$. For a differentiable, closed, and concave function g, its gradient is given by $\nabla g(\mathbf{u}) = \text{argmin}_{\mathbf{v}} \ \mathbf{v}^\top \mathbf{u} - g_*(\mathbf{v})$ and, furthermore, $g_{**} = g$. Again, with $\mathbf{v} = \nabla g(\mathbf{u})$, Fenchel-Young inequality with equality:

$$g(\mathbf{u}) + g_*(\nabla g(\mathbf{u})) = \mathbf{u}^\top \nabla g(\mathbf{u}). \tag{3}$$

3. Problem Statement

We formalize the problem statement described in Section 1 by explicitly describing the online and offline components, as well as the assumptions made on the payment functions of the customers and the procurement cost function of the seller.

As described in Section 1, had the seller known all the customers that were to arrive, they would have solved Problem (P-1) to obtain the optimal allocation to make to each customer. We denote the optimal value of Problem (P-1) as P^*. However, the challenge faced by the seller is that they have no knowledge of future customers, and so the seller must make decisions that trade off making a profit now with saving resources to potentially make a larger profit later. The seller knows the procurement cost function, f, before any customers arrive. Upon arrival, the customer reveals their payment function, v_t, and the seller must then make an irrevocable allocation before interacting with the next customer. In Section 7, we discuss an algorithm that does not require the customer to reveal their payment function. We have the following assumptions on the payment function of each customer.

Assumption 1 (Customer Payment Function). *The function $v_t : \mathbb{R}_+^D \to \mathbb{R}_+$ satisfies the following:*

1. *The function v_t is concave, differentiable, and closed.*
2. *The function v_t is increasing; that is, $\mathbf{u} \geq \mathbf{v}$ implies that $v_t(\mathbf{u}) \geq v_t(\mathbf{v})$.*
3. *The function v_t at $\vec{0}$ has value 0, that is, $v_t(\mathbf{0}) = 0$.*

Concavity in Assumption 1(1) comes from the idea that a customer is willing to pay marginally less for a larger bundle, which comes from the natural desire for the customer to receive a *bulk discount*. Assumption 1(2) reflects the customer's willingness to pay a larger amount for a larger bundle and Assumption 1(3) states that a customer would pay nothing for an empty bundle.

The procurement cost function satisfies the following assumptions.

Assumption 2 (Procurement Cost Function). *The function $f : \mathbb{R}_+^D \to \mathbb{R}_+$ satisfies the following:*

1. *The function f is convex, differentiable, and closed.*
2. *The function f is increasing; that is, $\mathbf{u} \geq \mathbf{v}$ implies that $f(\mathbf{u}) \geq f(\mathbf{v})$.*
3. *The function f at $\vec{0}$ has value 0, that is, $f(\mathbf{0}) = 0$.*

Convexity in Assumption 2(1) captures the idea that procuring scarce resources comes at an increasing cost. Assumption 2(2) comes from a larger cumulative allocation incurring a larger production cost and Assumption 2(3) states that the seller incurs no cost for allocating nothing.

3.1. Performance Metric

The performance of an algorithm making allocations in this setting is evaluated by its competitive ratio, which is the ratio of the objective value achieved by the algorithm to the offline optimum for all possible instances. We provide the formal definition below.

Definition 1 (Competitive Ratio). Consider the set of decision vectors produced by an algorithm, ALG, as $\{\bar{\mathbf{x}}_t\}_{t=1}^T$ and the offline optimal decision vector that achieves $P^\star$ from Problem (P-1) as $\{\mathbf{x}_t^\star\}_{t=1}^T$. Then, ALG has a competitive ratio of α if

$$\alpha \le \frac{\text{ALG}}{P^\star} = \frac{\sum_{t=1}^T v_t(\bar{\mathbf{x}}_t) - f(\sum_{t=1}^T \bar{\mathbf{x}}_t)}{\sum_{t=1}^T v_t(\mathbf{x}_t^\star) - f(\sum_{t=1}^T \mathbf{x}_t^\star)}$$

for all $\{v_t\}_{t=1}^T$.

Note that $\alpha \in [0,1]$ and the closer to one, the better the algorithm.

3.2. Primal-Dual Algorithm

We now present the primal-dual algorithm for the online optimization problem with procurement costs. We first express the dual of (P-1) as

$$D^\star := \underset{\boldsymbol{\lambda}\ge\mathbf{0},\, \mathbf{z}_t\ge\mathbf{0}}{\text{minimize}} \sum_{t=1}^T \sum_{d=1}^D \max\{[\mathbf{z}_t]_d - [\boldsymbol{\lambda}]_d, 0\} - \sum_{t=1}^T v_{t*}(\mathbf{z}_t) + f^*(\boldsymbol{\lambda}), \tag{D-1}$$

which is derived in Online Appendix A. The algorithm we develop solves an optimization problem for time t considering that decisions for time steps $[t-1]$ have already been made. Let $\bar{\mathbf{x}}_i$ denote the decision made by an algorithm at time i. The greedy solution at time t is the result of

$$\underset{\mathbf{0}\preceq\mathbf{x}_t\preceq\mathbf{1}}{\text{maximize}}\ v_t(\mathbf{x}_t) - f\left(\sum_{i=1}^{t-1}\bar{\mathbf{x}}_i + \mathbf{x}_t\right) + f\left(\sum_{i=1}^{t-1}\bar{\mathbf{x}}_i\right). \tag{M-1}$$

The objective of (M-1) represents the gain in the objective of (P-1) at time t if we make decision $\mathbf{x}_t$, because the decisions $\{\bar{\mathbf{x}}_i\}_{i=1}^{t-1}$ cannot be changed. From Assumption 2(1), we know that $f = f^{**}$, and from Assumption 1(1), we know that $v_t = v_{t**}$, which allows us to rewrite (M-1) as

$$\underset{\mathbf{0}\preceq\mathbf{x}_t\preceq\mathbf{1}}{\text{maximize}}\ \underset{\boldsymbol{\lambda}\ge\mathbf{0},\,\mathbf{z}_t\ge\mathbf{0}}{\text{minimize}}\ \mathbf{z}_t^\top\mathbf{x}_t - v_{t*}(\mathbf{z}_t) - \boldsymbol{\lambda}^\top\left(\sum_{i=1}^{t-1}\bar{\mathbf{x}}_i + \mathbf{x}_t\right) + f^*(\boldsymbol{\lambda}) + f\left(\sum_{i=1}^{t-1}\bar{\mathbf{x}}_i\right). \tag{M-2}$$

A greedy algorithm using this decision rule allocates at time t based on the incoming v_t, the previous decisions, and f. Unlike prior works of Chan et al. (2015), Azar et al. (2016), and Huang and Kim (2018), we do not require the procurement cost function to have a monotone increasing gradient (see Assumption 2); therefore, our framework captures more problems. To improve the performance of this algorithm, we ask the following question. *Can we design a surrogate function for f (with a monotone increasing gradient) such that decisions made with respect to this function give a better competitive ratio for our original problem?* Consider the following optimization problem, with the surrogate function denoted by f_s,

$$\begin{aligned} \text{maximize}\quad & \sum_{t=1}^T v_t(\mathbf{x}_t) - f_s\left(\sum_{t=1}^T \mathbf{x}_t\right) \\ \text{subject to}\quad & \mathbf{0}\preceq\mathbf{x}_t\preceq\mathbf{1} \quad \forall t\in[T], \end{aligned} \tag{P-2}$$

where $\mathbf{x}_t \in [0,1]^D$ is the optimization variable and $v_t : \mathbb{R}_+^D \to \mathbb{R}_+$ satisfies Assumption 1. The only difference between Problem (P-1) and Problem (P-2) is that f has been replaced by f_s, which satisfies the following assumptions.

Assumption 3 (Surrogate Function). *The function $f_s : \mathbb{R}_+^D \to \mathbb{R}_+$ satisfies the following:*

1. *The function f_s is convex, differentiable, and closed.*
2. *The function f_s is increasing; that is, $\mathbf{u}\ge\mathbf{v}$ implies $f_s(\mathbf{u})\ge f_s(\mathbf{v})$.*
3. *The function f_s at $\vec{0}$ has value 0, that is, $f_s(\mathbf{0}) = 0$.*
4. *The function f_s has an increasing gradient; that is, $\mathbf{u}\ge\mathbf{v}$ implies $\nabla f_s(\mathbf{u})\ge\nabla f_s(\mathbf{v})$.*
5. *The surrogate function is always larger than the procurement cost function, that is, $f_s(\mathbf{u})\ge f(\mathbf{u})$ for all $\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}$.*

Assumption 3, (1)–(3), is identical to Assumption 2, (1)–(3). Prior works, such as Azar et al. (2016), make Assumption 3(4) for the procurement cost function f. On the other hand, we relax this requirement and design a surrogate function f_s that satisfies this assumption instead. Assumption 3(5) is designed to make sure the resulting algorithm makes allocations more cautiously than the greedy algorithm without a surrogate function to best handle the uncertainty of future customers. Section 6 provides a simple example to illustrate this intuition. We discuss our choice of the surrogate function in more detail in Section 5.

Using the same strategy as above of writing the marginal optimization problem, now with respect to Problem (P-2), we can write the decision rule of Algorithm 1.

Algorithm 1 (Simultaneous Update)
Input: $f_s : \mathbb{R}^D \to \mathbb{R}$
1 **for** $t = 1, \ldots, T$ **do**
2 receive v_t;
3 $(\bar{\boldsymbol{\lambda}}_t, \bar{\mathbf{z}}_t, \bar{\mathbf{x}}_t) = \arg\min_{\boldsymbol{\lambda}\geq\mathbf{0}, \mathbf{z}_t\geq\mathbf{0}} \max_{\mathbf{0}\preceq\mathbf{x}_t\preceq\mathbf{s1}} \mathbf{z}_t^\top \mathbf{x}_t - v_{t*}(\mathbf{z}_t) - \boldsymbol{\lambda}^\top(\sum_{i=1}^{t-1}\bar{\mathbf{x}}_i + \mathbf{x}_t) + f_s^*(\boldsymbol{\lambda})$;

Line 3 in Algorithm 1, the main computational step of the algorithm, involves solving a (convex-concave) saddle-point problem. We point out that standard convex optimization methods (Bubeck 2015) can be used to solve this subproblem with desired accuracy, and the complexity analysis of these methods (number of iterations needed to reach ϵ-optimality) can be incorporated in the overall computational complexity analysis of our algorithm. In Section 7, we discuss an algorithm that computes the primal and dual variables sequentially.

In the remainder of this section, let $\bar{\mathbf{x}}_t$ denote the decision vector at time t given from Algorithm 1 called with f_s. Algorithm 1 called with f_s ensures that at every time step t,

$$\bar{\mathbf{x}}_t = \mathbf{1}[\bar{\mathbf{z}}_t - \bar{\boldsymbol{\lambda}}_t \geq \mathbf{0}], \tag{4}$$

where $\bar{\boldsymbol{\lambda}}_t = \nabla f_s(\sum_{i=1}^{t}\bar{\mathbf{x}}_i)$ and $\bar{\mathbf{z}}_t = \nabla v_t(\bar{\mathbf{x}}_t)$, which comes from the Karush–Kuhn–Tucker (KKT) conditions, as described in Online Appendix A.

The subscript notation of s—taken from *surrogate*—denotes the objective of Problem (P-1) resulting from the decision vectors coming from Algorithm 1 called with f_s. The primal objective is given by

$$P_s := \sum_{t=1}^{T} v_t(\bar{\mathbf{x}}_t) - f\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right), \tag{5}$$

and the dual objective is given by

$$D_s := \sum_{t=1}^{T}\sum_{d=1}^{D}\max\{[\bar{\mathbf{z}}_t]_d - [\bar{\boldsymbol{\lambda}}_t]_d, 0\} - \sum_{t=1}^{T} v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T). \tag{6}$$

These equations are used in the analysis of Algorithm 1 in Section 4.

4. Competitive Ratio Analysis for a Primal-Dual Algorithm

In this section, we bound the competitive ratio of Algorithm 1 called with f_s in Theorem 1. To do this, we first show that Algorithm 1 called with f_s does not make a decision that causes the objective to become negative.

Lemma 1 (Nonnegative Objective). *If f_s is convex and differentiable and $f_s(\mathbf{0}) = 0$, then*

$$\sum_{t=1}^{T} v_t(\bar{\mathbf{x}}_t) - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) \geq 0.$$

Proof. We upper bound this expression by incorporating the decision rule of Algorithm 1 called with f_s as follows:

$$\begin{aligned}
\sum_{t=1}^{T} v_t(\bar{\mathbf{x}}_t) - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) &\overset{(a)}{\geq} \sum_{t=1}^{T}\nabla v_t(\bar{\mathbf{x}}_t)^\top\bar{\mathbf{x}}_t - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) \\
&\overset{(b)}{=} \sum_{t=1}^{T}\left(\nabla v_t(\bar{\mathbf{x}}_t)^\top\bar{\mathbf{x}}_t - f_s\left(\sum_{i=1}^{t}\bar{\mathbf{x}}_i\right) + f_s\left(\sum_{i=1}^{t-1}\bar{\mathbf{x}}_i\right)\right) \\
&\overset{(c)}{\geq} \sum_{t=1}^{T}\left\langle\nabla v_t(\bar{\mathbf{x}}_t) - \nabla f_s\left(\sum_{i=1}^{t}\bar{\mathbf{x}}_i\right),\ \bar{\mathbf{x}}_t\right\rangle.
\end{aligned}$$

Inequality (a) comes from the concavity of v_t. Equality (b) comes from writing $f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)$ as a telescoping sum with the assumption that $f_s(\mathbf{0}) = 0$. Inequality (c) follows from convexity of f_s. Finally, the decision rule of Algorithm 1—that is, $\bar{\mathbf{x}}_t = \mathbf{1}[\bar{\mathbf{z}}_t - \bar{\boldsymbol{\lambda}}_t \succeq \mathbf{0}]$—called with f_s ensures that the inner product is always nonnegative. □

Now, we bound the competitive ratio of Algorithm 1 called with f_s.

Theorem 1 (Competitive Ratio Bound). *Suppose that $f_s : \mathbb{R}^D \to \mathbb{R}$ satisfies Assumption 3. The competitive ratio of Algorithm 1 (called with f_s) is bounded by $1/\alpha_{f,f_s}$, where*

$$\alpha_{f,f_s} := \sup_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u})}.$$

Proof. The general overview of the proof is as follows: writing D_s (6) in terms of P_s (5), we bound the gap between D_s and P_s. From here, we lower bound D_s by $D^\star$ (D-1), which in turn allows us to use weak duality to relate $D^\star$ and $P^\star$.

We start with writing D_s in terms of P_s:

$$\begin{aligned}
D_s &= \sum_{t=1}^{T}\sum_{d=1}^{D}\max\{[\bar{\mathbf{z}}_t]_d - [\bar{\boldsymbol{\lambda}}_t]_d, 0\} - \sum_{t=1}^{T} v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T) \\
&\overset{(a)}{=} \sum_{t=1}^{T}(\bar{\mathbf{z}}_t - \bar{\boldsymbol{\lambda}}_t)^\top\bar{\mathbf{x}}_t - \sum_{t=1}^{T} v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T) \\
&\overset{(b)}{=} \sum_{t=1}^{T}\nabla v_t(\bar{\mathbf{x}}_t)^\top\bar{\mathbf{x}}_t - \sum_{t=1}^{T}\nabla f_s\left(\sum_{i=1}^{t}\bar{\mathbf{x}}_i\right)^\top\bar{\mathbf{x}}_t - \sum_{t=1}^{T} v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T).
\end{aligned}$$

Equality (a) comes from the decision rule of Algorithm 1 called with f_s, which ensures that $\bar{\mathbf{x}}_t = \mathbf{1}[\bar{\mathbf{z}}_t - \bar{\boldsymbol{\lambda}}_t \succeq \mathbf{0}]$. Equality (b) comes from replacing $\bar{\boldsymbol{\lambda}}_t$ with $\nabla f_s(\sum_{i=1}^{t}\bar{\mathbf{x}}_i)$ and $\bar{\mathbf{z}}_t$ with $\nabla v_t(\bar{\mathbf{x}}_t)$. Now, we proceed to bound the duality gap between D_s and P_s by first observing the following relationship:

$$\begin{aligned}
D_s &\overset{(c)}{\le} \sum_{t=1}^{T}\nabla v_t(\bar{\mathbf{x}}_t)^\top\bar{\mathbf{x}}_t - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) - \sum_{t=1}^{T} v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T) \\
&\overset{(d)}{=} \sum_{t=1}^{T}\nabla v_t(\bar{\mathbf{x}}_t)^\top\bar{\mathbf{x}}_t - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) - \sum_{t=1}^{T}(\nabla v_t(\bar{\mathbf{x}}_t)^\top\bar{\mathbf{x}}_t - v_t(\bar{\mathbf{x}}_t)) + f^*(\bar{\boldsymbol{\lambda}}_T) \\
&= \sum_{t=1}^{T} v_t(\bar{\mathbf{x}}_t) - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) + f^*(\bar{\boldsymbol{\lambda}}_T) + f\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) - f\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) \\
&\overset{(e)}{=} P_s - f_s\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right) + f^*(\bar{\boldsymbol{\lambda}}_T) + f\left(\sum_{t=1}^{T}\bar{\mathbf{x}}_t\right).
\end{aligned}$$

Inequality (c) follows directly from the convexity of f_s. Equality (d) comes from the concave Fenchel-Young inequality—that is, Equation (3) with $g = v_t$ and $\mathbf{u} = \bar{\mathbf{x}}_t$. Equality (e) follows by substituting the definition of $P_s = \sum_{t=1}^{T} v_t(\bar{\mathbf{x}}_t) - f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)$ where in the preceding equality we add and subtract $f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)$. We bound the gap between D_s and P_s as a multiplicative factor of P_s to relate these quantities as a ratio:

$$\begin{aligned}
\frac{f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t) - f^*(\bar{\boldsymbol{\lambda}}_T) - f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)}{P_s} &\overset{(f)}{=} \frac{f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t) - f^*(\nabla f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)) - f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)}{\sum_{t=1}^{T} v_t(\bar{\mathbf{x}}_t) - f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)} \\
&\overset{(g)}{\ge} \frac{f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t) - f^*(\nabla f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)) - f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)}{f_s(\sum_{t=1}^{T}\bar{\mathbf{x}}_t) - f(\sum_{t=1}^{T}\bar{\mathbf{x}}_t)} \\
&\overset{(h)}{\ge} \inf_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{f_s(\mathbf{u}) - f^*(\nabla f_s(\mathbf{u})) - f(\mathbf{u})}{f_s(\mathbf{u}) - f(\mathbf{u})} =: \beta_{f,f_s}.
\end{aligned}$$

In equality (f), we replace $\bar{\boldsymbol{\lambda}}_T$ with $\nabla f_s(\sum_{i=1}^T \bar{\mathbf{x}}_i)$. Inequality (g) follows from Lemma 1 and the fact that the numerator is upper bounded by $P_s - D_s \le 0$ (see (e)) and hence nonpositive, and inequality (h) follows from observing that $\mathbf{0} \preceq \sum_{t=1}^T \bar{\mathbf{x}}_t \preceq T\mathbf{1}$. Hence,

$$\beta_{f,f_s} P_s \le f_s\left(\sum_{t=1}^T \bar{\mathbf{x}}_t\right) - f^*(\bar{\boldsymbol{\lambda}}_T) - f\left(\sum_{t=1}^T \bar{\mathbf{x}}_t\right) \le P_s - D_s.$$

Define

$$\alpha_{f,f_s} := 1 - \beta_{f,f_s} = \sup_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{f^*\left(\nabla f_s(\mathbf{u})\right)}{f_s(\mathbf{u}) - f(\mathbf{u})}. \tag{7}$$

Assumption 3(5) ensures that $\alpha_{f,f_s} \ge 0$, which, in turn, ensures that the competitive ratio is nonnegative and thus a meaningful quantity. We now lower bound D_s by $D^\star$ and subsequently use weak duality to get that $D_s \ge D^\star \ge P^\star$. From Assumption 3(3), we know that $\nabla f_s(\sum_{i=1}^t \bar{\mathbf{x}}_i) \preceq \nabla f_s(\sum_{i=1}^T \bar{\mathbf{x}}_i)$ for all $t \in [T]$ because $\bar{\mathbf{x}}_i \ge \mathbf{0}$ for all $i \in [T]$. This implies that $\bar{\boldsymbol{\lambda}}_t \preceq \bar{\boldsymbol{\lambda}}_T$ for all $t \in [T]$, so

$$\begin{aligned} D_s &= \sum_{t=1}^T \sum_{d=1}^D \max\{[\bar{\mathbf{z}}_t]_d - [\bar{\boldsymbol{\lambda}}_t]_d, 0\} - \sum_{t=1}^T v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T) \\ &\ge \sum_{t=1}^T \sum_{d=1}^D \max\{[\bar{\mathbf{z}}_t]_d - [\bar{\boldsymbol{\lambda}}_T]_d, 0\} - \sum_{t=1}^T v_{t*}(\bar{\mathbf{z}}_t) + f^*(\bar{\boldsymbol{\lambda}}_T) \ge D^\star. \end{aligned}$$

Hence, $P_s - D^\star \ge P_s \beta_{f,f_s}$, and applying weak duality, we get that $P_s - P^\star \ge P_s \beta_{f,f_s}$. Rearranging this equation gives us the following:

$$\frac{P_s}{P^\star} \ge \frac{1}{1 - \beta_{f,f_s}} = \frac{1}{\alpha_{f,f_s}}.$$

This concludes the proof. □

This theorem allows us to write the competitive ratio as the result of an optimization problem for a large class of f and f_s. Our objective then becomes to design f_s such that α_{f,f_s} is as small as possible because this would, in turn, yield a stronger competitive ratio bound. We can then verify the following intuition: To increase the denominator of (7), we see that we must craft f_s to be sufficiently larger than f to make cautious allocations in the face of adversarial uncertainty. However, to decrease the numerator of (7), we must not design f_s to be too large; otherwise, the algorithm will be overly cautious and make too little allocation. In the next section, we discuss how to choose f_s to optimize this ratio.

5. Designing the Surrogate Function

As the analysis in the preceding section shows, the choice of the surrogate function plays a crucial role in obtaining an improved competitive ratio bound. In this section, we propose techniques to design f_s for particular classes of functions. In particular, in Section 5.1, we propose a technique for designing the surrogate of polynomial functions and we obtain the competitive ratio bound in this setting. In Section 5.2, we exploit quasiconvex optimization to design the surrogate function for a general f.

5.1. Polynomial Function

We propose a design technique for a special class of f: polynomials that satisfy Assumption 2. We let $f_s(\mathbf{u}) = \frac{1}{\rho} f(\rho \mathbf{u}), \rho > 1$. Note that $\nabla f_s(\mathbf{u}) = \nabla f(\rho \mathbf{u})$. The intuition here is to stay cautious because we make no assumptions about the arriving input. Now, it suffices to determine ρ. This surrogate function was proposed in Chan et al. (2015), but their analysis yielded a suboptimal choice of ρ.

Theorem 2 shows that finding the optimal ρ for a general class of polynomial functions comes back to solving a one-dimensional optimization problem.

Theorem 2 (Competitive Ratio for Polynomial Functions). *Suppose that $\mathbf{u} \in \mathbb{R}^D$. For any $K \in \mathbb{N}$, suppose $f_K(\mathbf{u}) = \sum_{k=1}^K c_k g_k(\mathbf{u})$ is a convex function such that $c_k > 0$ for each $k \in [K]$ and $g_k(\mathbf{u}) = \prod_{i=1}^D u_i^{\tau_{ki}}$, where $\sum_{i=1}^D \tau_{ki} = \tau_k$, and $\tau_{ki} \in \mathbb{R}_+$ for all pairs $(k,i) \in [K] \times [D]$. Assume that $\tau := \tau_{i^*} \ge 2$, where $i^* = \arg\max_i \tau_i$. Then, choosing parameter ρ as $\rho = \tau^{1/(\tau-1)}$ guarantees a competitive ratio of at least $\tau^{-\tau/(\tau-1)}$ for Algorithm 1 called with $\frac{1}{\rho} f_K(\rho \mathbf{u})$.*

Proof. We first use induction to show that

$$\inf_{\rho>1} \sup_{\mathbf{0}\preceq \mathbf{u}\preceq T\mathbf{1}} \frac{f_K^*(\nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}))}{\frac{1}{\rho} f_K(\rho\mathbf{u}) - f_K(\mathbf{u})} \le \inf_{\rho>1} (\tau-1)\frac{\rho^\tau}{\rho^{\tau-1}-1},$$

and then apply Lemma 2 to the optimization problem. First note that for any $K \in \mathbb{N}$, the Fenchel-Young inequality holds with equality as described in Equation (2) in Section 2. That is,

$$f_K^*(\nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u})) = \langle \nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_K(\rho\mathbf{u}).$$

We now begin the inductive argument on K. For $K=1$, $f_1(\mathbf{u}) = c_1 \prod_{i=1}^D u_i^{\tau_{1,i}}$, where $\sum_{i=1}^D \tau_{1,i} = \tau_1 \ge 2$ and $\tau_{1,i}$ is non-negative for all i. Using the definition of f_1, we have

$$\begin{aligned}
\inf_{\rho>1} \sup_{\mathbf{0}\preceq \mathbf{u}\preceq T\mathbf{1}} \frac{f_1^*\left(\nabla_{\rho\mathbf{u}} f_1(\rho\mathbf{u})\right)}{\frac{1}{\rho} f_1(\rho\mathbf{u}) - f_1(\mathbf{u})} &\overset{(a)}{=} \inf_{\rho>1} \sup_{\mathbf{0}\preceq \mathbf{u}\preceq T\mathbf{1}} \frac{\langle \nabla_{\rho\mathbf{u}} f_1(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_1(\rho\mathbf{u})}{\frac{1}{\rho} f_1(\rho\mathbf{u}) - f_1(\mathbf{u})} \\
&\overset{(b)}{=} \inf_{\rho>1} \sup_{\mathbf{0}\preceq \mathbf{u}\preceq T\mathbf{1}} \frac{\rho^{\tau_1}(\tau_1-1) f_1(\mathbf{u})}{(\rho^{\tau_1-1}-1) f_1(\mathbf{u})} \\
&\overset{(c)}{=} \inf_{\rho>1} (\tau_1-1)\frac{\rho^{\tau_1}}{\rho^{\tau_1-1}-1}.
\end{aligned}$$

Equality (a) comes from the Fenchel-Young inequality holding with equality. Equality (b) comes from the following:

$$\begin{aligned}
f_1(\rho\mathbf{u}) &= \rho^{\tau_1} f_1(\mathbf{u}), \\
\langle \nabla_{\rho\mathbf{u}} f_1(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle &= \rho^{\tau_1}\tau_1 f_1(\mathbf{u}).
\end{aligned}$$

Equality (c) comes from removing $f_1(\mathbf{u})$ from the numerator and denominator, thus eliminating any dependence of $\mathbf{u}$ in the optimization problem. This concludes the proof for $K=1$.

Suppose that the result holds for $K-1 \in \mathbb{N}$. We argue the result for $K \in \mathbb{N}$. For notational convenience, we define

$$\begin{aligned}
a_k &:= c_k \rho^{\tau_k - \tau_K}(\tau_k - 1),\ b_k := c_k(\tau_K - 1)\left(\frac{\rho^{\tau_k-1}-1}{\rho^{\tau_K-1}-1}\right), \\
h_k(\mathbf{u}) &:= \left(g_k(\mathbf{u})\right)^{-1}.
\end{aligned}$$

Without loss of generality, let $\tau_1 \ge \cdots \ge \tau_K$ where $\tau_1 \ge 2$. We show that removing $c_K g_K(\mathbf{u})$ upper bounds the optimization problem. We begin with the following:

$$\begin{aligned}
\frac{\langle \nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_K(\rho\mathbf{u})}{\frac{1}{\rho} f_K(\rho\mathbf{u}) - f_K(\mathbf{u})} &\overset{(d)}{=} \frac{\sum_{k=1}^K c_k \rho^{\tau_k}(\tau_k-1) g_k(\mathbf{u})}{\sum_{k=1}^K c_k(\rho^{\tau_k-1}-1) g_k(\mathbf{u})} \\
&= \frac{\sum_{k=1}^{K-1} c_k \rho^{\tau_k}(\tau_k-1) g_k(\mathbf{u}) + c_K \rho^{\tau_K}(\tau_K-1) g_K(\mathbf{u})}{\sum_{k=1}^{K-1} c_k(\rho^{\tau_k-1}-1) g_k(\mathbf{u}) + c_K(\rho^{\tau_K-1}-1) g_K(\mathbf{u})} \\
&\overset{(e)}{=} \frac{\rho^{\tau_K}}{\rho^{\tau_K-1}-1}\left(\frac{h_K(\mathbf{u})\left(\sum_{k=1}^{K-1} a_k g_k(\mathbf{u})\right) + c_K(\tau_K-1)}{h_K(\mathbf{u})\left(\sum_{k=1}^{K-1} c_k\left(\frac{\rho^{\tau_k-1}-1}{\rho^{\tau_K-1}-1}\right) g_k(\mathbf{u})\right) + c_K}\right) \\
&\overset{(f)}{=} \frac{\rho^{\tau_K}}{\rho^{\tau_K-1}-1}\left((\tau_K-1) + \frac{h_K(\mathbf{u})\sum_{k=1}^{K-1}(a_k-b_k) g_k(\mathbf{u})}{h_K(\mathbf{u})\left(\sum_{k=1}^{K-1} c_k\left(\frac{\rho^{\tau_k-1}-1}{\rho^{\tau_K-1}-1}\right) g_k(\mathbf{u})\right) + c_K}\right).
\end{aligned}$$

Equality (d) comes from the following:

$$f_K(\rho\mathbf{u}) = \sum_{k=1}^{K} c_k \rho^{\tau_k} g_k(\mathbf{u}),$$
$$\langle \nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle = \sum_{k=1}^{K} \rho^{\tau_k}\tau_k g_k(\mathbf{u}).$$

Equality (e) comes from factoring out $\rho^{\tau_K} g_K(\mathbf{u})$ from the numerator and $(\rho^{\tau_K-1}-1)g_K(\mathbf{u})$ from the denominator. Equality (f) comes from rearranging the fraction inside the parentheses by bringing $(\tau_K - 1)$ out front.

$$a_k - b_k = c_k\left(\rho^{\tau_k-\tau_K}(\tau_k-1) - (\tau_K-1)\left(\frac{\rho^{\tau_k-1}-1}{\rho^{\tau_K-1}-1}\right)\right) \geq 0.$$

Now, we have

$$\begin{aligned}
\frac{\langle \nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_K(\rho\mathbf{u})}{\frac{1}{\rho}f_K(\rho\mathbf{u}) - f_K(\mathbf{u})} &\overset{(g)}{\leq} \frac{\rho^{\tau_K}}{\rho^{\tau_K-1}-1}\left((\tau_K-1) + \frac{h_K(\mathbf{u})\sum_{k=1}^{K-1}(a_k-b_k)g_k(\mathbf{u})}{h_K(\mathbf{u})\sum_{k=1}^{K-1} c_k\left(\frac{\rho^{\tau_k-1}-1}{\rho^{\tau_K-1}-1}\right)g_k(\mathbf{u})}\right)\\
&\overset{(h)}{=} \frac{\rho^{\tau_K}}{\rho^{\tau_K-1}-1}\left((\tau_K-1) + \frac{\sum_{k=1}^{K-1}(a_k-b_k)g_k(\mathbf{u})}{\sum_{k=1}^{K-1} c_k\left(\frac{\rho^{\tau_k-1}-1}{\rho^{\tau_K-1}-1}\right)g_k(\mathbf{u})}\right)\\
&\overset{(i)}{=} \frac{\sum_{k=1}^{K-1}\rho^{\tau_k}(\tau_k-1)g_k(\mathbf{u})}{\sum_{k=1}^{K-1}(\rho^{\tau_k-1}-1)g_k(\mathbf{u})}\\
&= \frac{\langle \nabla_{\rho\mathbf{u}} f_{K-1}(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_{K-1}(\rho\mathbf{u})}{\frac{1}{\rho}f_{K-1}(\rho\mathbf{u}) - f_{K-1}(\mathbf{u})}.
\end{aligned}$$

Inequality (g) comes from removing c_K from the denominator. Equality (h) comes from removing $h_K(\mathbf{u})$ from the numerator and denominator of the fraction inside the parentheses. Equality (i) comes from combining the expression back into a single fraction. We now finish the claim with the following:

$$\begin{aligned}
\inf_{\rho>1}\sup_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{f_K^*(\nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}))}{\frac{1}{\rho}f_K(\rho\mathbf{u}) - f_K(\mathbf{u})} &\overset{(j)}{=} \inf_{\rho>1}\sup_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{\langle \nabla_{\rho\mathbf{u}} f_K(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_K(\rho\mathbf{u})}{\frac{1}{\rho}f_K(\rho\mathbf{u}) - f_K(\mathbf{u})}\\
&\leq \inf_{\rho>1}\sup_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{\langle \nabla_{\rho\mathbf{u}} f_{K-1}(\rho\mathbf{u}),\ \rho\mathbf{u}\rangle - f_{K-1}(\rho\mathbf{u})}{\frac{1}{\rho}f_{K-1}(\rho\mathbf{u}) - f_{K-1}(\mathbf{u})}\\
&\overset{(k)}{=} \inf_{\rho>1}\sup_{\mathbf{0}\preceq\mathbf{u}\preceq T\mathbf{1}} \frac{f_{K-1}^*(\nabla_{\rho\mathbf{u}} f_{K-1}(\rho\mathbf{u}))}{\frac{1}{\rho}f_{K-1}(\rho\mathbf{u}) - f_{K-1}(\mathbf{u})}\\
&\overset{(l)}{\leq} \inf_{\rho>1}(\tau_1-1)\frac{\rho^{\tau_1}}{\rho^{\tau_1-1}-1}.
\end{aligned}$$

Equality (j) and equality (k) come from the Fenchel-Young inequality, which hold at equality. Inequality (l) comes from the inductive hypothesis.

We now apply Lemma 2 to solve

$$\arg\min_{\rho>1}(\tau_1-1)\frac{\rho^{\tau_1}}{\rho^{\tau_1-1}-1} = \tau_1^{1/(\tau_1-1)}.$$

Plugging this choice of ρ back into the objective gives us $\tau_1^{\tau_1/(\tau_1-1)}$ which concludes the proof. □

5.1.1. Comparison with Chan et al. (2015). Chan et al. (2015) approach a similar optimization problem but exploit their additional assumptions on the procurement cost function that essentially restricts their class to polynomials. They choose their design parameter to be $\rho = \lambda^{1/(\lambda-1)}$, where λ is defined as the smallest cumulative degree of a term in f; that is, $\lambda := \tau_{j^\star}$, where $j^\star = \arg\min_j \tau_j$. Chan et al. (2015) are interested in the asymptotic behavior of the

competitive ratio in terms of τ, and both their choice of ρ and our choice of ρ give the same $\mathcal{O}(\tau)$ competitive ratio bound.[1] However, we achieve a more refined competitive ratio bound with our choice of $\rho = \tau^{1/(\tau-1)}$.

5.2. General Case

In this section, we propose a design approach for a general procurement cost function. We show that the algorithm metric we aim to optimize is a quasiconvex function of f_s, the surrogate function we are aiming to design. Therefore, the search over an appropriate family of f_s can be carried out by quasiconvex optimization. Note that, although the approach is general, solving the problem computationally requires discretizing the variable $\mathbf{u} \in \mathbb{R}_+^D$, and thus this method is suitable for cases where D is small.

Theorem 3 (Surrogate Function Design). *Let* $\mathbf{a} \in \mathbb{R}_+^N$, *where* $\mathbf{1}^\top \mathbf{a} \geq 1$, *and* $f_s(\mathbf{u}) = \sum_{n=1}^N a_n g_n(\mathbf{u})$, *where* g_n *satisfies Assumption 3 for all* $n \in [N]$. *Consider a discretization of the set* $\{\mathbf{u} | \mathbf{0} \preceq \mathbf{u} \preceq T\mathbf{1}\}$ *and denote the points in this discretized set as* $\mathcal{U}$. *The following problem*

$$\underset{\mathbf{a} \geq \mathbf{1}}{\text{minimize}} \ \max_{\mathbf{u} \in \mathcal{U}} \frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u})} \tag{Q-1}$$

can be solved as a quasiconvex optimization problem.

Proof. To show that Problem (Q-1) is a quasiconvex optimization problem, we must verify that the constraints are convex and the objective is quasiconvex. It suffices to show that

$$\max_{\mathbf{u} \in \mathcal{U}} \frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u})}$$

is a quasiconvex function in $\mathbf{a}$. Because a nonnegative weighted maximum of quasiconvex functions is also quasiconvex, it suffices to show that $\frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u})}$ is quasiconvex in $\mathbf{a}$ for a fixed $\mathbf{u}$. We can directly apply the definition of quasiconvexity. Let $S_\alpha(f_s)$ be the sublevel sets of f_s for $\mathbf{a} \in \mathbb{R}_+^N$. We have the following:

$$\begin{aligned} S_\alpha(f_s) &= \left\{ \mathbf{1}^\top \mathbf{a} \geq 1 \,\middle|\, \frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u})} \leq \alpha \right\} \\ &= \{ \mathbf{1}^\top \mathbf{a} \geq 1 | f^*(\nabla f_s(\mathbf{u})) \leq \alpha (f_s(\mathbf{u}) - f(\mathbf{u})) \}. \end{aligned}$$

For a fixed value of $\mathbf{u}$, $[\nabla f_s(\mathbf{u})]_d$ is linear in $\mathbf{a}$ for all d, and because f^* is always convex, composing a convex function with a linear function of $\mathbf{a}$ is convex in $\mathbf{a}$. Finally, because $f_s(\mathbf{u})$ is linear in $\mathbf{a}$, the constraints of $S_\alpha(f_s)$ are convex, and thus $S_\alpha(f_s)$ is a convex set. □

Because Problem (Q-1) is a quasiconvex optimization problem from Theorem 3, we can solve it by a sequence of convex feasibility problems, using bisection on α; see Online Appendix C for details and psuedocode.

6. Numerical Examples

In this section, we illustrate the performance of our algorithm for specific procurement cost functions. In our first example, we use a simple procurement cost function to demonstrate the need for a surrogate function when calling Algorithm 1. In our second example, we consider a nonseparable polynomial procurement cost function and compare the performance of Algorithm 1 for different surrogate function design techniques.

Example 1. Consider the procurement cost function $f(u) = u^2$, where $u \in \mathbb{R}_+$. The following numerical example shows the necessity of a surrogate function, and how running Algorithm 1 with the original procurement cost function has a competitive ratio of zero asymptotically. We show this by crafting a particular arrival instance in which we highlight the weakness of not using a surrogate function. The intuition is that not using a surrogate function allows the decision making to be excessively greedy, in that the algorithm does not caution itself from accumulating a large procurement cost for minimal revenue. In this instance, the incoming valuations are linear, and so we have $v_t(x_t) = c_t x_t$. We have $c_t = 2t$. Assume that T is divisible by two. From the decision rule of Algorithm 1 called with f_s, that is, $\overline{x}_t = \mathbf{1}[c_t - f_s'(\sum_{i=1}^t \overline{x}_i)]$, calling Algorithm 1 with $f_s(u) = u^2$ leads to an allocation of $x_t = 1$ for all t which gives a cumulative reward of T. The optimal allocation is one that sets $x_t = 1$ for all $t > \frac{T}{2}$ and yields an objective of $\frac{1}{2}(T^2 + T)$. Therefore, using $f_s(u) = u^2$, Algorithm 1 returns a set of decisions that has a competitive ratio of zero as T becomes large. Both of our design techniques give a surrogate function of $f_s(u) = 2u^2$. Calling Algorithm 1 with $f_s(u) = 2u^2$ leads to an allocation of $x_t = 0.5$ for all t, which gives an objective of $\frac{T^2}{4} + \frac{T}{2}$. Therefore, using $f_s(u) = 2u^2$, Algorithm 1 returns a set of decisions that has a competitive ratio of $\frac{1}{2}$ as T becomes large. This example then shows that not using a surrogate function may lead to a competitive ratio that tends to zero as T becomes large.

Example 2. Now, consider the procurement cost function $f(\mathbf{u}) = u_1^4 + (u_1 + u_2)^2$, where $\mathbf{u} \in \mathbb{R}^2_+$. Figure 1 shows the shape of the surrogate function using the design techniques from Sections 5.1 and 5.2, respectively. For f_s from Section 5.1, we use the surrogate function $f_s(\mathbf{u}) = \frac{1}{\rho} f(\rho \mathbf{u})$, and with Theorem 2, we choose $\rho = 4^{1/3}$. This means that $f_s(\mathbf{u}) = 4u_1^4 + 4^{1/3}(u_1 + u_2)^2$. This choice of ρ then gives a competitive ratio bound of $4^{-4/3} \approx 0.1575$. For f_s from Section 5.2, we use surrogate function $f_s(\mathbf{u}) = a_1 u_1^4 + a_2(u_1 + u_2)^2$ from Theorem 3. To solve Problem (Q-1), we set $T = 10$ and have 100 points per square unit in the discretization; that is,

$$\mathcal{U} = \{\mathbf{u} \mid u_i \in \{0, 0.1, 0.2, \ldots, 9.9, 10\} \quad \forall i \in \{1, 2\}\}.$$

This achieves the competitive ratio bound of approximately 0.1577 with $a_1 \approx 3.791$ and $a_2 \approx 2.386$. The surrogate function from Section 5.2 allows for an additional design parameter that allows us to achieve a slightly better competitive ratio bound than the surrogate function from Section 5.1. However, the technique from Section 5.2 has a much higher computational cost due to numerically solving the quasiconvex optimization problem in Problem (Q-1). Figure 2 compares the cumulative objective values up to time t, that is, $\sum_{i=1}^{t} v_i(\bar{\mathbf{x}}_i) - f(\sum_{i=1}^{t} \bar{\mathbf{x}}_i)$, of Algorithm 1 called with different surrogate functions. For the surrogate functions, we have the label f representing the surrogate function equal to the original production cost function, and so Algorithm 1 is called with $u_1^4 + (u_1 + u_2)^2$. We have the label f_{poly} representing the surrogate function from using the technique in Section 5.1, so Algorithm 1 is called with $4u_1^4 + 4^{1/3}(u_1 + u_2)^2$. We have the label f_{design} representing the surrogate function from using the technique in Section 5.2, so Algorithm 1 is called with $a_1 u_1^4 + a_2(u_1 + u_2)^2$ where $a_1 \approx 3.791$ and $a_2 \approx 2.386$. Finally, we have the label f_{chk} representing the surrogate function from using the technique in Chan et al. (2015), so Algorithm 1 is called with $8u_1^3 + 2(u_1 + u_2)^2$. The online arrivals are generated by reasoning about instances that would be adversarial for Algorithm 1 called with the original procurement cost function, that is, arrivals that would cause the algorithm to behave too greedily and amass a large procurement cost for minimal revenue. In this instance, the incoming valuations are linear, that is, $v_t(\mathbf{x}_t) = \mathbf{c}_t^\top \mathbf{x}_t$, where

$$\mathbf{c}_t = \begin{cases} \nabla f(t \cdot \mathbf{1}) & \text{if } t \text{ is odd} \\ \nabla f(2t \cdot \mathbf{1}) & \text{if } t \text{ is even.} \end{cases}$$

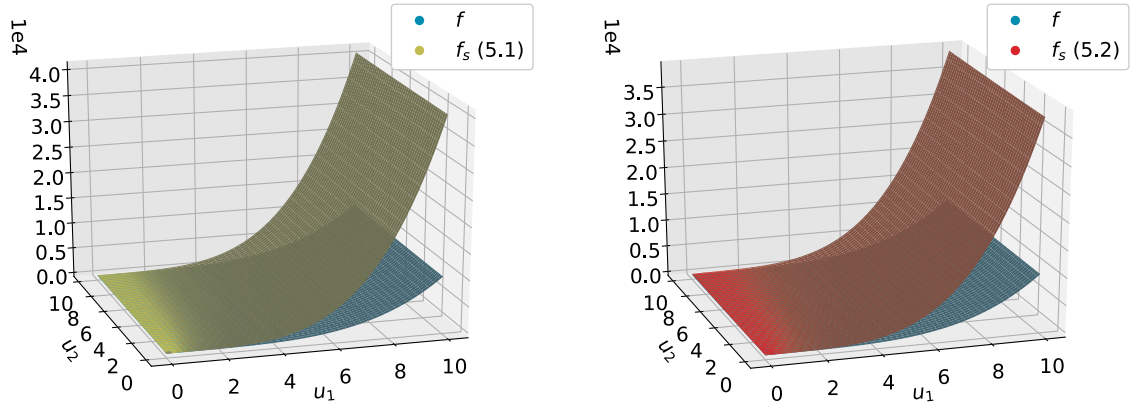
7. Posted Pricing Mechanisms

In this section, we propose Algorithm 2, which is a primal-dual algorithm that computes the primal and dual variables sequentially, unlike Algorithm 1, which computes the primal and dual variables simultaneously as the solution to the saddle-point problem in Equation (M-2). Algorithm 2 is much more computationally efficient and possesses an economic interpretation of *incentive compatibility* as defined in Definition 2.

Definition 2 (Incentive Compatibility). An online algorithm for Problem (P-1) is called incentive compatible when each customer maximizes their utility by being truthful, that is, each customer reports and acts according to their true beliefs.

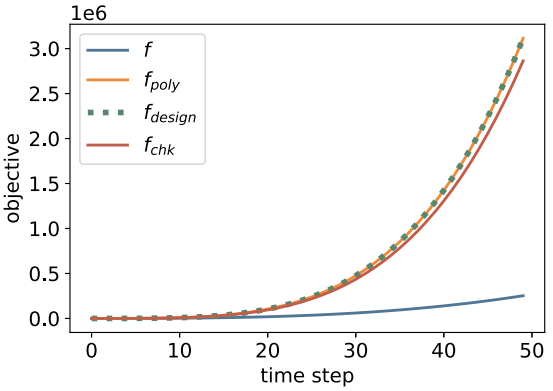
Algorithm 2 is called a posted pricing mechanism, as defined in Definition 3, and immediately satisfies *incentive compatibility*.

Figure 1. (Color online) Problem (Q-1) Applied to $f(\mathbf{u}) = u_1^4 + (u_1 + u_2)^2$



Note. The surrogate functions corresponding to Section 5.1 (left) and Section 5.2 (right) are shown.

Figure 2. (Color online) Plot Comparing the Objectives up to Time t of Algorithm 1 Called with Different Surrogate Functions



Note. For the surrogate functions, we have f representing the surrogate function equal to f, f_{poly} representing the surrogate function from using the technique in Section 5.1, f_{design} representing the surrogate function from using the technique in Section 5.2, and f_{chk} using the technique in Chan et al. (2015).

Definition 3 (Posted Pricing Mechanism). An online algorithm is a posted pricing mechanism when the seller posts item prices and allows the arriving customer to choose their desired bundle of items given the prices.

The interpretation here is that upon arrival, the customer chooses the allocation that maximizes their utility, and this would be identical to the allocation that the seller would assign had the user reported their true valuation function. From the notation of Algorithm 2, the dual variable, $\bar{\boldsymbol{\lambda}}_t$, represents a price that is revealed at each time step, *before* the customer arrives, and then the allocation for this arriving customer is then determined by this price. The posted price at time step t, therefore, does not depend on v_t, so the arriving agent does not need to reveal it. A posted pricing mechanism is therefore desirable in applications where the privacy of v_t is important.

Algorithm 2 (Sequential Update with Offset)
Input: $f : \mathbb{R}^D \to \mathbb{R}, \mathbf{v}_{\text{offset}} \in \mathbb{R}^D_+$
for $t = 1 \ldots T$ **do**
  $\bar{\boldsymbol{\lambda}}_t = \nabla f_s(\sum_{i=1}^{t-1} \bar{\mathbf{x}}_i + \mathbf{v}_{\text{offset}})$;
  $\bar{\mathbf{x}}_t = \arg\max_{\mathbf{0} \preceq \mathbf{x}_t \preceq \mathbf{1}} v_t(\mathbf{x}_t) - \bar{\boldsymbol{\lambda}}_t^\top \mathbf{x}_t$

We propose the primal-dual algorithm in Algorithm 2. Here, in comparison with Algorithm 1, $\bar{\boldsymbol{\lambda}}_t$ is being used to set the threshold at time t, independent of the allocation made at time t. Thus, the value of $\bar{\boldsymbol{\lambda}}_t$ does not require solving a saddle-point problem. Furthermore, in comparison with Algorithm 1, in addition to passing in the function, f, as an argument, we pass in an offset vector, $\mathbf{v}_{\text{offset}}$, to Algorithm 2 that allows us to additively control the threshold. The naming of both Algorithm 1 as Simultaneous Update and Algorithm 2 as Sequential Update to distinguish between how the primal and dual variables are computed comes from Eghbali and Fazel (2016).

7.1. Analysis Without Offset

In this section, we analyze the competitive ratio of Algorithm 2 called with $\mathbf{v}_{\text{offset}} = \mathbf{0}$ and f_s satisfying Assumption 3. This ensures that at every time step t, $\bar{\mathbf{x}}_t = \mathbf{1}[\bar{\mathbf{z}}_t - \bar{\boldsymbol{\lambda}}_t \geq \mathbf{0}]$, where $\bar{\boldsymbol{\lambda}}_t = \nabla f_s(\sum_{i=1}^{t-1} \bar{\mathbf{x}}_i)$ and $\mathbf{z}_t = \nabla v_t(\bar{\mathbf{x}}_t)$ from Lemma 4. Now, we bound the competitive ratio of Algorithm 2.

Theorem 4 (Competitive Ratio Without Offset). *Let f_s satisfy Assumption 3. The competitive ratio of Algorithm 2 called with f_s and $\mathbf{v}_{\text{offset}} = \mathbf{0}$ is bounded by $1/\alpha_{f,f_s}$ where*

$$\alpha_{f,f_s} := \sup_{\mathbf{0} \preceq \mathbf{u} \preceq T\mathbf{1}} \frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u}) - \mathbf{1}^\top(\nabla f_s(\mathbf{u}) - \nabla f_s(\mathbf{0}))}.$$

This proof is very similar to that of Theorem 1 and so the proof is provided in Online Appendix D.1.

7.1.1. Designing the General Surrogate Function.

In a similar vein to Section 5.2, we now propose a design technique for the surrogate function, f_s to be used in Algorithm 2 based on Theorem 4.

Theorem 5 (Surrogate Function Design Without Offset). *Let* $f(\mathbf{u}) = \sum_{n=1}^{N} g_n(\mathbf{u})$ *where* g_n *satisfies Assumption 2 for all* $n \in [N]$. *Let* $\mathbf{a} \in \mathbb{R}^N$, *where* $\mathbf{a} \geq \mathbf{1}$, *and* $f_s(\mathbf{u}) = \sum_{n=1}^{N} a_n g_n(\mathbf{u})$. *Consider a discretization of the set* $\{\mathbf{u} | \mathbf{0} \preceq \mathbf{u} \preceq T\mathbf{1}\}$ *and denote the points in this discretized set as* $\mathcal{U}$. *The following problem*

$$\underset{\mathbf{a} \geq \mathbf{1}}{\text{minimize}} \max_{\mathbf{u} \in \mathcal{U}} \frac{f^*(\nabla f_s(\mathbf{u}))}{f_s(\mathbf{u}) - f(\mathbf{u}) - \mathbf{1}^\top(\nabla f_s(\mathbf{u}) - \nabla f_s(\mathbf{0}))} \tag{Q-2}$$

can be solved as a quasiconvex optimization problem.

This proof is very similar to that of Theorem 3 and so the proof is provided in Online Appendix D.2.

7.2. Analysis with Offset

In this section, we show that posting a *more cautious* price, that is, setting a larger threshold due to the uncertainty from the allocation, allows for a clean analysis of the competitive ratio of Algorithm 2. We term a larger price as *more cautious* because an allocation is not made unless the larger threshold is reached, implying a larger degree of caution for the current time step. This larger threshold comes from the assumption that the gradient of f_s is increasing, and so adding a nonnegative offset to the argument increases ∇f_s.

In this section, we analyze Algorithm 2 called with f_s satisfying Assumption 4 and $\mathbf{v}_{\text{offset}} = \mathbf{1}$. This ensures that at every time step t, $\bar{\mathbf{x}}_t = \mathbf{1}[\bar{\mathbf{z}}_t - \bar{\boldsymbol{\lambda}}_t \geq \mathbf{0}]$, where $\bar{\boldsymbol{\lambda}}_t = \nabla f_s(\sum_{i=1}^{t-1} \bar{\mathbf{x}}_i + \mathbf{1})$ and $\bar{\mathbf{z}}_t = v_t(\bar{\mathbf{x}}_t)$ from Lemma 4.

We now consider the following assumptions on f_s.

Assumption 4 (Surrogate Function). *The function* $f_s : \mathbb{R}_+^D \to \mathbb{R}_+$ *satisfies the following:*

1. *The function* f_s *is convex, differentiable, and closed.*
2. *The function* f_s *is increasing; that is,* $\mathbf{u} \geq \mathbf{v}$ *implies* $f_s(\mathbf{u}) \geq f_s(\mathbf{v})$.
3. *The function* f_s *at* $\vec{0}$ *has value* 0, *that is,* f_s *has an increasing gradient; that is,* $\mathbf{u} \geq \mathbf{v}$ *implies* $\nabla f_s(\mathbf{u}) \geq f_s(\mathbf{v})$.
4. *The function* $f_s(\mathbf{0}) = 0$.
5. *The surrogate function is always larger than the procurement cost function, that is,* $f_s(\mathbf{u}) \geq f(\mathbf{u})$ *for all* $\mathbf{0} \preceq \mathbf{u} \preceq (T-1)\mathbf{1}$.
6. *The following holds:* $f_s(\mathbf{a}) - f(\mathbf{a}) \leq f_s(\mathbf{b}) - f(\mathbf{b})$ *if* $\mathbf{0} \preceq \mathbf{a} \preceq \mathbf{b}$.

Assumption 4, (1)–(4), is identical to Assumption 3, (1)–(4).

We now bound the competitive ratio of Algorithm 2.

Theorem 6 (Competitive Ratio with Offset). *Let* f_s *satisfy Assumption 4. The competitive ratio of Algorithm 2 called with* f_s *and* $\mathbf{v}_{\text{offset}} = \mathbf{1}$ *is bounded by* $1/\alpha_{f,f_s}$ *where*

$$\alpha_{f,f_s} := \sup_{\mathbf{0} \preceq \mathbf{u} \preceq (T-1)\mathbf{1}} \frac{f^*(\nabla f_s(\mathbf{u}+\mathbf{1}))}{f_s(\mathbf{u}) - f(\mathbf{u})}.$$

This proof is very similar to that of Theorem 1 and so the proof is provided in Online Appendix D.3.

7.2.1. Designing the General Surrogate Function.

In a similar vein to Section 5.2, we now propose a design technique for the surrogate function, f_s to be used in Algorithm 2 based on Theorem 6.

Theorem 7 (Surrogate Function Design with Offset). *Let* $f(\mathbf{u}) = \sum_{n=1}^{N} g_n(\mathbf{u})$ *where* g_n *satisfies Assumption 2 for all* $n \in [N]$. *Let* $\mathbf{a} \in \mathbb{R}^N$, *where* $\mathbf{a} \geq \mathbf{1}$, *and* $f_s(\mathbf{u}) = \sum_{n=1}^{N} a_n g_n(\mathbf{u})$. *Consider a discretization of the set* $\{\mathbf{u} | \mathbf{0} \preceq \mathbf{u} \preceq (T-1)\mathbf{1}\}$ *and denote the points in this discretized set as* $\mathcal{U}$. *The following problem*

$$\underset{\mathbf{a} \geq \mathbf{1}}{\text{minimize}} \max_{\mathbf{u} \in \mathcal{U}} \frac{f^*(\nabla f_s(\mathbf{u}+\mathbf{1}))}{f_s(\mathbf{u}) - f(\mathbf{u})} \tag{Q-3}$$

can be solved as a quasiconvex optimization problem.

This proof is very similar to that of Theorem 3 and is provided in Online Appendix D.4.

8. Related Work

In this section, we review further related work at the intersection of online matching and combinatorial auctions.

8.1. Online Bipartite Matching

Online bipartite matching discussed in Karp et al. (1990), Kalyanasundaram and Pruhs (2000), Devanur et al. (2013), and Kesselheim et al. (2013), among other works, is a classical problem that has been studied and reintroduced for many applications. Recently, the natural application of Internet ad placement has caused a resurgence

of online bipartite matching and its generalizations through the AdWords problem as seen in Mehta et al. (2007) and Devanur and Hayes (2009). In the AdWords problem, a search engine is trying to maximize revenue from a set of budget-constrained advertisers, who bid on queries arriving online. The AdWords problem is different from our framework in the following respects. The first aspect is that the AdWords problem is not a special case of our setting with the linear objective function. The AdWords problem could be rewritten as maximizing $\sum_{t=1}^{T} b_t^T x_t - f(\sum_{t=1}^{T} \text{diag}(b_t)x_t)$ subject to $0 \preceq x_t \preceq 1, 1^T x_t \leq 1 \ \ \forall t \in [T]$ where $[b_t]_i$ is the bid of advertiser i (with budget B_i) for the tth impression, $\text{diag}(b_t)$ is a diagonal matrix with b_t on its diagonal and $f(z) = 0$ if $z_i \leq B_i \ \ \forall i$ and infinity otherwise. However, in our framework, the online arriving information (e.g., b_t) does not appear in the argument of f. The second aspect is that in the AdWords problem, there is a hard budget constraint for each advertiser, and if we enforce this constraint in a penalized fashion (as above), the penalty function f would not be differentiable. On the other hand, in our setting, we consider a soft budget where additional resources could be acquired with cost according to a prespecified differentiable procurement cost function f. Devanur and Jain (2012) generalized the AdWords problem to allow the revenue to be the sum of a concave function of the budget spent for each advertiser. All the aforementioned problems have a separable cumulative budget constraint that must be satisfied, and so the algorithm techniques of choosing the allocation as a function of the budget are not applicable to our problem.

8.2. Primal-Dual Algorithms

State-of-the-art techniques for AdWords, its generalizations, and related problems have been primal-dual algorithms as discussed in Buchbinder et al. (2007) and Buchbinder and Naor (2009). A primal-dual algorithm uses the dual problem formulation and updates the dual variables to determine the values of the primal variables. The advantages of primal-dual algorithms are two-fold. Firstly, the analysis for the competitive ratio of a primal-dual algorithm then decomposes into writing the dual objective of the algorithm in terms of the primal objective, because weak duality can then be used to connect the two (see the opening paragraph of our proof of Theorem 1). Secondly, the dual variable may have a meaningful interpretation of how to determine the primal variable. We adopt the intuition for primal and dual variables from problems of profit maximization as in Balcan et al. (2008) and Chawla et al. (2010). Although these problems are different from our framework, Balcan et al. (2008) considers a limited or unlimited supply of resources, and Chawla et al. (2010) considers customers arriving from a known distribution, the interpretations of the primal and dual variables are key in developing our posted pricing mechanism in Section 7. In both Algorithm 1 and Algorithm 2, our allocation rules come naturally from realizing that the payment obtained must be greater than the additional production cost. The dual variable can then be interpreted as the price offered to the incoming buyer, as further discussed in Section 7.

This powerful tool of duality is best seen in online covering and packing problems Chan et al. (2015) and Azar et al. (2016). The offline covering problem can be written as

$$\underset{\mathbf{x}\in\mathbb{R}_+^n}{\text{minimize}} \ f(\mathbf{x}) \quad \text{subject to } A\mathbf{x} \geq \mathbf{1},$$

where f is a nonnegative increasing convex cost function and A is an $m \times n$ matrix with nonnegative entries. In the online problem, rows of A come online and a feasible assignment $\mathbf{x}$ must be maintained at all times where $\mathbf{x}$ may only increase. The offline covering problem can be written as

$$\underset{\mathbf{y}\in\mathbb{R}_+^m}{\text{maximize}} \ \sum_j y_j - f^*(A^\top \mathbf{y}),$$

and in the online setting, columns of $A^\top$ arrive online upon which y_t must be assigned. The packing problem is dual to the covering problem as the jth entry of $\mathbf{y}$ corresponds to the jth row of A. In the works of Chan et al. (2015) and Azar et al. (2016), the authors use this duality to analyze similar algorithms proposed for each problem. The bulk of the results in Chan et al. (2015) are focused on the covering and packing problems, upon which the authors then adapt their results to the online resource allocation problem in Section 5 of their work. In this paper, we obtain stronger results for the online resource allocation problem by studying the problem directly rather than trying to adapt results from the related problem of online packing.

We share a similar perspective in this work with Eghbali and Fazel (2016). The authors there study a generalization of AdWords in which the objective is a concave function and constraint sets and linear maps arrive online and propose a convex optimization problem to design a surrogate function to improve the competitive ratio. However, the problem studied in Eghbali and Fazel (2016) is different from ours in the following ways: (1) the data coming online in Eghbali and Fazel (2016) is linear, whereas in our setting the payment functions arriving online are generally concave, and (2) the objective of the offline optimization problem in Eghbali and Fazel

(2016) is a coupled term between allocations at different rounds, but in our objective, in Equation (P-1), we have a sum over decoupled terms representing the cumulative payment, as well as a coupled term in the procurement cost function. Because these key differences do not allow our problem to be mapped to that in Eghbali and Fazel (2016), we must develop separate surrogate function design techniques based on the competitive ratio analysis for our problem.

8.3. Arrival Models

Most of the online optimization problems analyzed with respect to competitive ratio are studied under three arrival models: (1) the worst-case/adversarial model, with no assumptions on how the requests arrive; (2) the random order model, where the set of requests is arbitrary but the order of arrival is uniformly random; and (3) the independently and identically distributed (IID) model, where the requests are IID samples from an underlying distribution. For a more in-depth survey, see section 2.2 in Mehta (2013). Our setting is that of the worst-case model. The key approach to problems in the worst-case model is for the decision maker to apply a greedy algorithm that maximizes a function of how much revenue can be immediately gained versus how much revenue may be achieved later. In doing so, the decision maker must be cautious in spending the budget or accumulating a resource that may be better consumed in the future. This decision-making strategy connects loosely to the ideas of regularization for online optimization problems in the regret metric as seen in classical algorithms such as follow the regularized leader, as discussed in McMahan (2011), and multiplicative weights, introduced in Littlestone and Warmuth (1994). A key difference however from the regret setting to the competitive ratio setting is that in the regret setting, regularization aims to keep the gap between the current and previous decision small, whereas, in the competitive ratio setting, regularization is used to make cautious decisions to protect resources that may obtain more value if used in future allocations.

In the random order model, the typical approach is to have an exploration period, where the decision maker learns about the distribution of the arriving requests, followed by an exploitation period in which the decision maker uses this knowledge to maximize their revenue. This is most clearly seen in the classical secretary problem described in Chow et al. (1964) in which a set of candidates arrive one by one for an open job position, and the manager must hire or reject the candidate before interviewing future candidates. AdWords is studied in the random order model in Devanur and Hayes (2009) and the algorithm proposed uses the same technique of initial exploration, in which the bids on the first few queries are used to learn weights on the bidders used to select the allocation, and an exploitation period, in which these weights are applied to future queries to make the assignment. Similar strategies are used for generalizations of AdWords such as online linear programming, as in Agrawal et al. (2014) and Agrawal and Devanur (2014), and profit maximization subject to convex costs, as in Gupta et al. (2018). The key difference between the random order model and our setting of the worst-case model is that previous customers tell us nothing about future customers, and so we forgo learning about our customers and focus solely on cautiously allocating our resources.

8.4. Online Combinatorial Auctions

In many related works, our problem of online resource allocation has been titled *online combinatorial auctions*. Online combinatorial auctions have been studied in the setting with fixed resource capacities; that is, there is a hard budget constraint for each resource, as discussed in Blumrosen and Nisan (2007), Balcan et al. (2008), Chakraborty et al. (2013), and Tan et al. (2020), and in the setting with unlimited resource supplies, in which additional resources can be acquired at no cost, such as Balcan et al. (2005) and Balcan et al. (2008). Our setting falls in between these; resources can be acquired or developed following a procurement cost. This problem was proposed by Blum et al. (2011) for separable procurement cost functions in the worst-case arrival model. Blum et al. (2011) devised a posted pricing mechanism, in which customers wanting to purchase the kth copy of any item would be charged a price equal to the procurement cost of the $2k$th copy of that item. Huang and Kim (2018) build on this result by characterizing the competitive ratio of optimal algorithms in this setting for a wide range of separable procurement costs as the solution to a differential equation. Our framework generalizes this setting by considering nonseparable production cost functions. Additionally, we bring an optimization viewpoint to this setting in which we use (quasi-)convex optimization to design the best surrogate function, rather than restricting ourselves to a small function class as do these papers.

9. Conclusion and Future Directions

In this paper, we studied the broad online optimization framework of online resource allocation with procurement costs. We analyzed the competitive ratio for a primal-dual algorithm and showed how we can design a

surrogate function to improve the competitive ratio. We proposed two techniques to design or shape the surrogate function. The first technique, discussed in Section 5.1, addressed the case of polynomial cost functions and determined a closed-form choice for the scalar design parameter, that guarantees a competitive ratio of at least $\tau^{-\tau/(\tau-1)}$, where τ is the largest cumulative degree of a single term in the polynomial. This bound is optimal from a result in Huang and Kim (2018) (theorem 10). The second technique, discussed in Section 5.2, considered a general class of procurement cost functions and relied on an optimization problem, which is quasiconvex in the design parameters, to determine a surrogate function. This allowed us to further improve the competitive ratio at a higher computational cost. In Section 6 we investigated the surrogate function arising from each design technique for numerical examples.

As a future direction, we aim to generalize Theorem 3 to allow a much larger class of functions for the design of the surrogate. We will also investigate which choice of g_n would lead to optimal smoothing for a certain class of f. Future steps also include a modified analysis that would allow more flexibility in f but make more assumptions on the arriving inputs. Additionally, practically motivated assumptions on the structure of the incoming payment functions might lead to competitive ratio results for Algorithm 1 that will not approach zero if f_s is close to f. Furthermore, the different assumptions on the input order such as the random order model may be more suitable for certain applications, and competitive analysis in this regime has yet to be studied for this exact problem. In addition, different assumptions on the procurement cost function may be better suited for applications where the procurement cost functions satisfy the increasing gradient property, that is, Assumption 2(3) (continuous supermodular functions), but are not necessarily convex as discussed in Sadeghi and Fazel (2020), Sadeghi et al. (2020a, b, 2021), and Raut et al. (2021).

Acknowledgments

M. Ray and O. Sadeghi contributed equally to this work.

Endnote

[1] In their work, Chan et al. (2015) define the competitive ratio to be the inverse of ours; to avoid confusion in case the reader refers to their work, we compare their result with ours according to their definition of competitive ratio.

References

Agrawal S, Devanur NR (2014) Fast algorithms for online stochastic convex programming. *Proc. 26th Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 1405–1424.

Agrawal S, Wang Z, Ye Y (2014) A dynamic near-optimal algorithm for online linear programming. *Oper. Res.* 62(4):876–890.

Andrews M, Antonakopoulos S, Zhang L (2016) Minimum-cost network design with (dis)economies of scale. *SIAM J. Comput.* 45(1):49–66.

Azar Y, Buchbinder N, Chan TH, Chen S, Cohen IR, Gupta A, Huang Z, et al. (2016) Online algorithms for covering and packing problems with convex objectives. *Proc. Annual IEEE Sympos. Foundations Comput. Sci.* (IEEE, Piscataway, NJ), 148–157.

Balcan MF, Blum A, Mansour Y (2008) Item pricing for revenue maximization. *Proc. 9th ACM Conf. Electronic Commerce* (ACM, New York), 50–59.

Balcan MF, Blum A, Hartline JD, Mansour Y (2005) Mechanism design via machine learning. *Proc. 46th Annual IEEE Sympos. Foundations Comput. Sci* (IEEE, New York), 605–614.

Bartal Y, Gonen R, Nisan N (2003) Incentive compatible multi unit combinatorial auctions. *Proc. 9th Conf. Theoretical Aspects Rationality Knowledge* (ACM, New York), 72–87.

Bertsimas D, Hawkins J, Perakis G (2009) Optimal bidding in online auctions. *J. Revenue Pricing Management* 8(1):21–41.

Blatter M, Muehlemann S, Schenker S (2012) The costs of hiring skilled workers. *Eur. Econom. Rev.* 56(1):20–35.

Blum A, Gupta A, Mansour Y, Sharma A (2011) Welfare and profit maximization with production costs. *Proc. Annual IEEE Sympos. Foundations Comput. Sci.* (IEEE, Piscataway, NJ).

Blumrosen L, Nisan N (2007) Combinatorial auctions. *Algorithmic Game Theory* (Cambridge University Press, Cambridge, UK), 267–300.

Bubeck S (2015) Convex optimization: Algorithms and complexity. *Foundations Trends Machine Learn.* 8(3–4):231–357.

Buchbinder N, Naor JS (2009) The design of competitive online algorithms via a primal–dual approach. *Foundations Trends Theoretical Comput. Sci.* 3:93–263.

Buchbinder N, Jain K, Naor JS (2007) Online primal-dual algorithms for maximizing ad-auctions revenue. *Proc. Eur. Sympos. Algorithms* (Springer, Berlin), 253–264.

Chakraborty T, Huang Z, Khanna S (2013) Dynamic and nonuniform pricing strategies for revenue maximization. *SIAM J. Comput.* 42(6):2424–2451.

Chan T, Huang Z, Kang N (2015) Online convex covering and packing problems. Preprint, submitted February 6, https://arxiv.org/abs/1502.01802.

Chawla S, Hartline JD, Malec DL, Sivan B (2010) Multi-parameter mechanism design and sequential posted pricing. *Proc. 42nd ACM Sympos. Theory Comput.* (ACM, New York), 311–320.

Chow Y, Moriguti S, Robbins H, Samuels S (1964) Optimal selection based on relative rank (the "secretary problem"). *Israel J. Math.* 2(2):81–90.

Devanur NR, Hayes TP (2009) The adwords problem: Online keyword matching with budgeted bidders under random permutations. *Proc. 10th ACM Conf. Electronic Commerce* (ACM, New York), 71–78.

Devanur NR, Jain K (2012) Online matching with concave returns. *Proc. 44th Annual ACM Sympos. Theory Comput.* (ACM, New York), 137–144.

Devanur NR, Jain K, Kleinberg RD (2013) Randomized primal-dual analysis of ranking for online bipartite matching. *Proc. 24th Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 101–107.

Eghbali R, Fazel M (2016) *Worst Case Competitive Analysis for Online Conic Optimization* (Neural Information Processing Systems).

Erdogan SA, Gose A, Denton BT (2015) Online appointment sequencing and scheduling. *IIE Trans.* 47(11):1267–1286.

Gupta A, Mehta R, Molinaro M (2018) Maximizing profit with convex costs in the random-order model. *Proc. 45th Internat. Colloquium Automata Languages Programming*, Leibniz International Proceedings in Informatics, vol. 107, 71:1–71:14.

Ho CJ, Vaughan JW (2012) Online task assignment in crowdsourcing markets. *Proc. 26th AAAI Conf. Artificial Intelligence* (AAAI Press, Cambridge, MA).

Huang Z, Kim A (2018) Welfare maximization with production costs: A primal dual approach. *Games Econom. Behav.* 1:1–20.

Hwang D, Jaillet P, Manshadi V (2021) Online resource allocation under partially predictable demand. *Oper. Res.* 69(3):895–915.

Jaillet P, Lu X (2012) Near-optimal online algorithms for dynamic resource allocation problems. Preprint, submitted August 13, https://arxiv.org/abs/1208.2596.

Kalyanasundaram B, Pruhs KR (2000) An optimal deterministic algorithm for online b-matching. *Theoretical Comput. Sci.* 233(1–2):319–325.

Karp RM, Vazirani UV, Vazirani VV (1990) An optimal algorithm for on-line bipartite matching. *Proc. 22nd Annual ACM Sympos. Theory Comput.* (ACM, New York), 352–358.

Kesselheim T, Radke K, Tönnis A, Vöcking B (2013) An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. *Proc. Eur. Sympos. Algorithms* (Springer, Berlin), 589–600.

Legrain A, Jaillet P (2013) Stochastic online bipartite resource allocation problems. Technical report, CIRRELT, Canada.

Littlestone N, Warmuth MK (1994) The weighted majority algorithm. *Inform. Comput.* 108(2):212–261.

Makarychev K, Sviridenko M (2014) Solving optimization problems with diseconomies of scale via decoupling. *Proc. IEEE 55th Annual Sympos. Foundations Comput. Sci.* (IEEE, New York), 571–580.

McMahan B (2011) Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. *Proc. 14th Internat. Conf. Artificial Intelligence Statist.* (PMLR, New York), 525–533.

Mehta A (2013) Online matching and ad allocation. *Foundations Trends Theoretical Comput. Sci.* 8(4):265–368.

Mehta A, Vazirani U, Vazirani V (2007) AdWords and generalized online matching. *J. ACM* 54(5):19.

Raut P, Sadeghi O, Fazel M (2021) Online DR-submodular maximization: Minimizing regret and constraint violation. *Proc. Conf. AAAI Artificial Intelligence* 35:9395–9402.

Sadeghi O, Fazel M (2020) Online continuous DR-submodular maximization with long-term budget constraints. *Proc. Internat. Conf. Artificial Intelligence Statist.* (PMLR, New York), 4410–4419.

Sadeghi O, Eghbali R, Fazel M (2020a) Online algorithms for budget-constrained dr-submodular maximization. *Proc. ICML Workshop Negative Dependence Submodularity ML* (MIT Press, Cambridge, MA).

Sadeghi O, Raut P, Fazel M (2020b) A single recipe for online submodular maximization with adversarial or stochastic constraints. *Adv. Neural Inform. Processing Systems* 33:14712–14723.

Sadeghi O, Raut P, Fazel M (2021) Improved regret bounds for online submodular maximization. *Proc. ICML Workshop Subset Selection Machine Learn.: From Theory to Applications* (ICML, San Diego).

Tan X, Sun B, Leon-Garcia A, Wu Y, Tsang DH (2020) Mechanism design for online resource allocation: A unified approach. *Proc. ACM Measurement Analysis Comput. Systems*, vol. 4 (ACM, New York), 1–46.