Optimal Dynamic Auctions are Virtual Welfare Maximizers*

Vahab Mirrokni† Renato Paes Leme† Pingzhong Tang‡
Song Zuo†
†Google Research, {mirrokni,renatoppl,szuo}@google.com
‡Tsinghua University, kenshinping@gmail.com

Abstract

We are interested in the setting where a seller sells sequentially arriving items, one per period, via a *dynamic auction*. At the beginning of each period, each buyer draws a private valuation for the item to be sold in that period and this valuation is independent across buyers and periods. The auction can be dynamic in the sense that the auction at period t can be conditional on the bids in that period and all previous periods, subject to certain appropriately defined *incentive compatible* and *individually rational* conditions. Perhaps not surprisingly, the revenue optimal dynamic auctions are computationally hard to find and existing literatures that aim to approximate the optimal auctions are all based on solving complex dynamic programs. It remains largely open on the structural interpretability of the optimal dynamic auctions.

In this paper, we show that any optimal dynamic auction is a virtual welfare maximizer subject to some monotone allocation constraints. In particular, the explicit definition of the virtual value function above arises naturally from the primal-dual analysis by relaxing the monotone constraints. We further develop an ironing technique that gets rid of the monotone allocation constraints. Quite different from Myerson's ironing approach, our technique is more technically involved due to the interdependence of the virtual value functions across buyers. We nevertheless show that ironing can be done approximately and efficiently, which in turn leads to a Fully Polynomial Time Approximation Scheme of the optimal dynamic auction.

*The authors thank the anonymous reviewers for their helpful comments.

Electronic copy available at: https://ssrn.com/abstract=3282600

1 Introduction

Recently, the problem of designing multi-period dynamic mechanisms has been shown to be both theoretically challenging and practically important. In particular, when running a sequence of repeated auctions on online advertising platforms, using dynamic auctions optimized across different time periods could potentially bring significant gains both in terms of revenue and social welfare. The power of dynamic mechanisms has been investigated by a number of recent papers [Bergemann and Välimäki, 2002; Parkes and Singh, 2004; Cavallo, 2008; Athey and Segal, 2013; Kakade *et al.*, 2013; Pai and Vohra, 2013; Pavan *et al.*, 2014; Devanur *et al.*, 2015; Balseiro *et al.*, 2016; Chawla *et al.*, 2016; Bergemann *et al.*, 2017; Balseiro *et al.*, 2017a,b; Lobel and Paes Leme, 2017; Shen *et al.*, 2018; Balseiro *et al.*, 2019]. We refer to Bergemann and Said [2011] and Bergemann and Välimäki [2017] for comprehensive surveys on the subject.

In particular, we consider a setting where a seller repeatedly interacts with a set of buyers and sells one item per period. The value of each buyer for the item at each period are independently drawn from commonly known prior distributions (no need to be identical) at the beginning of that period. The seller is allowed to sell the item of each period not only depending on the bids submitted in the current period, but also the histories, i.e., all the bids submitted in past periods. In the meanwhile, the seller must guarantee the dynamic auctions to be *ex-post individual rational* — the cumulative utility for each buyer is always positive, and *dynamic incentive compatible* — bidding truthfully (i.e., submitting private values as bids) to the auction is optimal for each buyer by taking into consideration the effect of current bids on future outcomes.

Even though dynamic mechanisms can be much more effective in maximizing revenue and social welfare [Jackson and Sonnenschein, 2007; Papadimitriou *et al.*, 2016], they have not been widely adopted in practice. The main issue of implementing dynamic mechanisms in practice is their high complexity. The complexity induced by the exponentially growing design space makes it difficult to solve or even to describe such mechanisms.

A series of recent work has made progress to resolve the complexity issue described above. For example, Ashlagi *et al.* [2016] and Mirrokni *et al.* [2018b] show that it is enough for the optimal dynamic auction to depend on scalar summaries of histories instead of the full history. However, the approximately optimal mechanisms described in these papers are solutions of complex dynamic programs that are written into a large table. It is therefore very difficult to understand the structure of these mechanisms. It remains open whether there is an intuitive structural characterization of the optimal dynamic auction.

In this paper, we show that the exact optimal dynamic auction has a very simple structural interpretation: the optimal dynamic auction in each period is a second price auction on a certain appropriately defined virtual value space.[1] More specifically, such virtual values (before ironing[2][3]) are quite similar to Myerson's virtual value [Myerson, 1981], i.e., they have the form of linear combinations of private values and Myerson's virtual values (before ironing). However, just like Myerson's auction, to make the virtual welfare[4] maximizing allocation rule monotone, one need to first iron the virtual values. Unlike the ironing in Myerson's auction, the ironing step in our case is interdependent across the values of different buyers. In other words, one's virtual value after ironing

[1] A virtual value function is a map from buyer value space to real numbers. A virtual value is the corresponding real number of some private value of the buyer.

[2] Informally, the ironing of a virtual value function is an operation that maps the virtual value function to another virtual value function (we call an ironed virtual value function) while preserving the expectation. For the formal definition, see Definition 3.1.

[3] Some recent works on virtual value and ironing [Elkind, 2007; Roughgarden and Schrijvers, 2016].

[4] Virtual welfare is the sum of virtual values of the buyers who get the item. For expected virtual welfare, the expectation is taken over the randomness from the allocation rules.

Electronic copy available at: https://ssrn.com/abstract=3282600

not only depends on his/her own value, but also on other buyers' values. Although the ironing step here is not as simple as the ironing in Myerson's auction, its computation is still efficient for constant many buyer cases. Moreover, we provide a Fully Polynomial Time Approximation Scheme to compute the virtual values for any period of the dynamic auction given the histories so far, and such virtual values induce a dynamic auction with revenue arbitrarily close to the optimal.

Techniques There are two main techniques used in our analysis for optimal dynamic auctions. The first is the so-called *bank account mechanisms* which are a subset of dynamic auctions with simple structures and can achieve the optimal revenue of all the dynamic auctions [Mirrokni *et al.*, 2016a,c, 2018a,b]. Briefly speaking, a bank account mechanism keeps a state for each buyer as the summary of the history of each buyer, which is a scalar called *balance*. Each period depends on the previous periods through the vector of buyer balances. With the bank account framework, the designer only needs to specify single-period auctions that are single-period incentive compatible together with a valid balance update policy. In other words, the design of an entire dynamic auction breaks into the design of a series of single-period auctions and a balance update policy. The decomposition greatly simplifies the problem and enables clean mathematical programs for each period.

The second is a primal-dual analysis and a sensitivity analysis of the parametric mathematical programs. In fact, primal-dual analysis is commonly used in economic studies [Nisan *et al.*, 2007, Chapter 5, Daskalakis *et al.*, 2013; Cai *et al.*, 2016; Cai and Zhao, 2017; Daskalakis *et al.*, 2017]. In particular, for our problem, we can prove that these programs are convex and satisfy the Slater's conditions. Hence the solution is optimal if and only if the KKT conditions are satisfied. From these conditions, we show that the auction in each period maximizes some virtual welfare and we also discover the exact form of the virtual values with partial relaxation on the monotonicity constraints of allocation. Furthermore, we show that adding these relaxed constraints back to the program indeed corresponds to ironing the virtual values. As we mentioned previously, the ironing step here is interdependent across the values of the buyers and hence different from the ironing step in Myerson's auction. To resolve this difficulty, we show a method to algorithmically accomplish the ironing step.

2 Preliminaries

We study a setting where a seller repeatedly interacts with k buyers selling one item per period over T periods. The value of each buyer $i \in [k]$ for the item in period $t \in [T]$ is $v_i^t \in \mathcal{V}$. If $x_i^t \in [0,1]$ represents the probability that buyer i is allocated the item in period t, his utility is $v_i^t \cdot x_i^t$. Throughout this paper, (i) we use subscripts as the indices of buyers, bold fonts for vectors of all the buyers (i.e., $\mathbf{a} = (a_1, \ldots, a_k)$), and subscript $-i$ for the vector except the i-th element (i.e., $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$); (ii) we use superscripts as the indices of periods and $a^{1..t}$ to denote the sequence $a^1, \ldots, a^t$.

The values v_i^t are assumed to be drawn from independent distributions F_i^t.[5][6] The distributions F_i^t are assumed to be common knowledge but the realizations of the random variables are initially unknown for both the buyers and the designer. At each period, the following events happen:

1. each buyer i learns his value $v_i^t \sim F_i^t$;

[5] In practice, it is still fair to assume that the values of the buyers (advertisers) are independent conditional on each particular inventory and cookie.

[6] If the distributions are not independent, then the weak version of truthfulness still holds, i.e., truthful if all other buyers bid truthfully.

Electronic copy available at: https://ssrn.com/abstract=3282600

2. each buyer i reports value $\hat{v}_i^t$ to the designer;
3. the designer implements an outcome $\boldsymbol{x}^t \in [0,1]^k$ and charges the buyers $\boldsymbol{p}^t$;
4. each buyer accrues utility $u_i^t = v_i^t \cdot x_i^t - p_i^t$.

A dynamic mechanism can then be described in terms of a pair of maps for each period, which associate the history of reports $\hat{\boldsymbol{v}}^{1..t} = (\hat{\boldsymbol{v}}^1, \hat{\boldsymbol{v}}^2, \ldots, \hat{\boldsymbol{v}}^t)$ to an outcome $\boldsymbol{x}^t$ and payment $\boldsymbol{p}^t$:

$$\text{Outcome: } \boldsymbol{x}^t : \mathcal{V}^{kt} \to [0,1]^k \qquad \text{Payment: } \boldsymbol{p}^t : \mathcal{V}^{kt} \to \mathbb{R}^k.$$

Therefore we can define: $u_i^t(v_i^t; \hat{\boldsymbol{v}}^{1..t}) = v_i^t \cdot x_i^t(\hat{\boldsymbol{v}}^{1..t}) - p_i^t(\hat{\boldsymbol{v}}^{1..t})$.

2.1 Dynamic Incentive Compatibility

A mechanism is incentive compatible if it provides incentives for buyers to reveal their true types in each iteration. Such conditions for dynamic mechanisms can be easily defined by backward induction: in the last period, regardless of the history so far and other buyers' reports, it should be incentive compatible for each buyer to report his true value. This corresponds to the usual notion of incentive compatibility in (static) mechanism design:

$$v_i^T = \operatorname{argmax}_{\hat{v}_i^T} u_i^T(v_i^T; \hat{\boldsymbol{v}}^{1..T})$$

for all $i, \hat{\boldsymbol{v}}^{1..T-1}, \hat{v}_{-i}^T, v_i^T$. To simplify notations, from now on we will omit the 'for-all' quantification and assume all expressions are quantified as 'for-all' in its free variables. For the next-to-last-period, it should be incentive compatible for the buyer to report his true value given that he will report his true value in the following period:

$$v_i^{T-1} = \operatorname{argmax}_{\hat{v}_i^{T-1}} u_i^{T-1}(v_i^{T-1}; \hat{\boldsymbol{v}}^{1..T-1}) + \mathbb{E}_{v_i^T}[u_i^T(v_i^T; \hat{\boldsymbol{v}}^{1..T-1}, v_i^T, \hat{v}_{-i}^T)].$$

Proceeding by backward induction, we require that:

$$v_i^t = \operatorname{argmax}_{\hat{v}_i^t} u_i^t(v_i^t; \hat{\boldsymbol{v}}^{1..t}) + U_i^t(\hat{v}_i^{1..t} | \hat{v}_{-i}^{1..T}) \tag{DIC}$$

where the second term is the *continuation utility*, i.e., the expected utility obtained from the subsequent periods of the mechanism assuming the buyer reports truthfully:

$$U_i^t(\hat{v}_i^{1..t} | \hat{v}_{-i}^{1..T}) := \mathbb{E}_{v_i^{t+1..T}}\left[\textstyle\sum_{\tau=t+1}^T u_i^\tau(v_i^\tau; \hat{\boldsymbol{v}}^{1..t}, v_i^{t+1..\tau}, \hat{v}_{-i}^{t+1..\tau})\right]$$

A well-known fact in dynamic mechanism design is that DIC implies that buyer i's expected overall utility $U_i^0(\hat{v}_{-i}^{1..T})$ is maximized by reporting truthfully in each period.

2.2 Ex-Post Individual Rationality

Another desirable constraint is ex-post individual rationality which says that a buyer should derive non-negative utility from the mechanism for every realization of the values:

$$\textstyle\sum_{t=1}^T u_i^t(v_i^t; \boldsymbol{v}^{1..t}) \geq 0 \tag{eP-IR}$$

We focus on the problem of maximizing revenue subject to DIC, eP-IR, and feasibility constraints:

$$\begin{aligned} \max \quad & \textsc{Rev} = \mathbb{E}[\textstyle\sum_{t=1}^T \sum_{i=1}^k p_i^t(\boldsymbol{v}^{1..t})] \\ \text{s.t.} \quad & \text{(DIC), (eP-IR), and feasibility: } \textstyle\sum_{i=1}^k x_i^t(\boldsymbol{v}^{1..t}) \leq 1 \end{aligned}$$

Electronic copy available at: https://ssrn.com/abstract=3282600

2.3 Bank Account Mechanisms

The space of mechanisms satisfying DIC and eP-IR is very broad and unstructured. We restrict our attention in this section to a subclass of dynamic mechanisms introduced by Mirrokni *et al.* [2016a] called *bank account mechanisms*. The mechanisms are simple, dynamic incentive compatible by design and have the following notable features:

Lemma 2.1 (Mirrokni *et al.* [2016b])**.** *Given any dynamic mechanism satisfying DIC and eP-IR, there exists a bank account mechanism with at least the same revenue and at least the same welfare.*

In particular, for any given setting, there is a revenue-optimal mechanism in the form of a bank account mechanism.

Bank account mechanisms keep a state for each buyer, which is a scalar called *balance*. Each period depends on the previous periods through the vector of buyer balances. Another main feature is that in this framework, the designer needs to specify single-period auctions that are single-period incentive compatible together with a valid balance update policy. That is, once a valid balance update policy is in place, all the designer needs to worry about are single-period incentive compatibility constraints.

A bank account mechanism B is defined in terms of the following functions for each period:

- A static single-period auction $\boldsymbol{x}^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}), \boldsymbol{p}^{B,t}(\boldsymbol{v}^t, \boldsymbol{b})$ parameterized by a balance vector $\boldsymbol{b} \in \mathbb{R}_+^k$ that is (single-period) *incentive-compatible* for each $\boldsymbol{b}$, i.e.:

$$v_i^t \cdot x_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}) - p_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}) \geq v_i^t \cdot x_i^{B,t}(\hat{v}_i^t, v_{-i}^t, \boldsymbol{b}) - p_i^{B,t}(\hat{v}_i^t, v_{-i}^t, \boldsymbol{b}) \tag{IC}$$

- Note that we do not require the mechanism to be (single-period) individually rational. We also require the utility of the buyer to be *balance independent in expectation*, i.e., that:

$$\mathop{\mathbb{E}}_{v_i^t \sim F_i^t}\left[v_i^t \cdot x_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}) - p_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b})\right] \text{ is a non-negative constant not depending on } \boldsymbol{b} \tag{BI}$$

- A balance update policy $\boldsymbol{b}^{B,t}(\boldsymbol{v}^t, \boldsymbol{b})$ which maps the previous balances and the reports to the current balances, satisfying the following *balance update* conditions:

$$0 \leq b_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}) \leq b^i + v_i^t \cdot x_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}) - p_i^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}) \tag{BU}$$

  Given the balance update functions, we can define $\boldsymbol{b}^t : \mathcal{V}^t \to \mathbb{R}_+^k$ recursively as:

$$\boldsymbol{b}^0 = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{b}^1(\boldsymbol{v}^1) = \boldsymbol{b}^{B,1}(\boldsymbol{v}^1, \boldsymbol{0}) \quad \text{and} \quad \boldsymbol{b}^t(\boldsymbol{v}^{1..t}) = \boldsymbol{b}^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}^{B,t-1}(\boldsymbol{v}^{1..t-1}))$$

  which allows us to define a dynamic mechanism in the standard sense as:

$$\boldsymbol{x}^t(\boldsymbol{v}^{1..t}) = \boldsymbol{x}^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}^{t-1}), \quad \boldsymbol{p}^t(\boldsymbol{v}^{1..t}) = \boldsymbol{p}^{B,t}(\boldsymbol{v}^t, \boldsymbol{b}^{t-1}).$$

In what follows we will abuse notations by dropping the superscript B and refer to $\boldsymbol{x}^t(\boldsymbol{v}^{1..t})$ and $\boldsymbol{x}^t(\boldsymbol{v}^t, \boldsymbol{b}^{t-1})$ interchangeably. One important theorem from previous studies is that any bank account mechanism satisfies stronger notions of DIC and eP-IR.

Lemma 2.2 (Mirrokni *et al.* [2016b])**.** *Any bank account mechanism satisfying IC, BI, and BU is DIC and eP-IR.*

Electronic copy available at: https://ssrn.com/abstract=3282600

3 The Structure of Optimal Bank Account Mechanisms

In this section, we show our first main result that identifies the underlying structure of the optimal bank account mechanism (hence also the optimal dynamic auctions). Interestingly, the dynamic auctions with the structure can be interpreted as an ironed virtual welfare maximizing auction, where the ironed virtual value is defined as follows.

Definition 3.1 (Ironing)**.** *The ironing operation on the virtual value functions of all buyers transfers these virtual value functions into some other virtual value functions (called ironed virtual value functions) while preserving the following properties:*

- Monotonicity*: The allocation rule that maximizes the ironed virtual welfare is monotone, i.e., for each buyer i, the allocation probability is weakly increasing in v_i for any fixed v_{-i}.*
- Limited transfer*: The expectation of virtual value conditional on each allocation equivalence class is unchanged. An allocation equivalence class of buyer i with any given v_{-i} is a maximal subset of his/her private values where the allocation probability to buyer i is a constant. Again, the allocation rule here is the one that maximizes the ironed virtual welfare.*
- First order dominance*: The ironed virtual value is first order dominated by the virtual value before ironing.*

Informally, we have the following theorem:

Theorem 3.2 (Informal)**.** *Any revenue optimal bank account mechanism is maximizing some virtual value (after ironing) for each period.*

We will restate the formal version of this theorem after introducing all necessary notations (Theorem 3.8). In particular, the specific form of the virtual value (before ironing) will be provided.

3.1 Formalizing the subprogram for each period

We start with the subproblem of optimizing period t while all other periods are fixed. In particular, the problem can be formalized as a convex program.

Lemma 3.3 (Convex program)**.** *For each period t, any valid balance $\boldsymbol{b}^t$, and fixed expected utility for period t, if mechanism is fixed for the remaining periods ($t+1$ to T), then the optimal auction for period t can be solved via a convex program.*

To proof this statement smoothly, we show how to formalize the program step by step in the rest of this subsection rather than putting everything into a single proof environment. For the ease of presentation, we will hide the superscript ^{t} while focusing on a single period.

Let $\boldsymbol{b}$ be the bank account balance given at the beginning of this period and

$$\xi_i(v_{-i}) := \mathbb{E}_{v_i}[v_i \cdot x(\boldsymbol{v}, \boldsymbol{b}) - p(\boldsymbol{v}, \boldsymbol{b})] \tag{1}$$

be the expected utility of buyer i in this period. Note that by (BI), it is independent of the bank account balance $\boldsymbol{b}$, while it could be different for different bids from other buyers, i.e., v_{-i}. In fact, the ξ are some parameters we will need to determine later based on the distribution of $\boldsymbol{b}$ induced by the auctions in each period.

Electronic copy available at: https://ssrn.com/abstract=3282600

Objective The expected revenue acquired within this period can be computed by the expected social welfare minus the expected utility of all the buyers,

$$\textsc{PeriodRev}(\boldsymbol{b}, \xi) = \mathbb{E}_{\boldsymbol{v}}\left[\sum\nolimits_i (v_i \cdot x_i(\boldsymbol{v}, \boldsymbol{b}) - \xi_i(v_{-i}))\right].$$

Besides the expected revenue from the current period, we also need to include the expected revenue for future periods in the objective. We then use $g(\boldsymbol{b}'|M)$ to denote the expected total revenue the seller can collect from all upcoming periods by following a given bank account mechanism M in these periods, where $\boldsymbol{b}'$ is the vector of the bank account balance at the end of the current period. In fact, $\boldsymbol{b}' = \boldsymbol{b} + \Delta\boldsymbol{b}(\boldsymbol{v})$ is a function of buyers' bids $\boldsymbol{v}$ in this period, where $\Delta\boldsymbol{b}(\boldsymbol{v})$ are the changes in the bank accounts.

Then the expected revenue since current period is,

$$\textsc{Rev}(\boldsymbol{b}, \xi) = \mathbb{E}_{\boldsymbol{v}}\left[\sum\nolimits_i (v_i \cdot x_i(\boldsymbol{v}, \boldsymbol{b}) - \xi_i(v_{-i})) + g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b}|M)\right]. \tag{2}$$

Constraints Now we proceed to the constraints of the optimization. According to Lemma 2.2, the mechanism must satisfy constraint (IC), (BI), and (BU). Note that (BI) is guaranteed by the definition of $\xi_i(v_{-i})$, we then formalize constraint (IC) and (BU).

For (IC), as we won't introduce the payment variables, it is enough to require the monotonicity for the allocation rules only, i.e.,

$$\text{(IC)} \iff x_i(\boldsymbol{v}, \boldsymbol{b}) \text{ is monotone in } v_i \text{ for fixed } v_{-i}. \tag{3}$$

Since by the Envelope theorem [Rochet, 1985],

$$\frac{\partial(v_i \cdot x_i(\boldsymbol{v}, \boldsymbol{b}) - p_i(\boldsymbol{v}, \boldsymbol{b}))}{\partial v_i} = x_i(\boldsymbol{v}, \boldsymbol{b}) \implies u_i(\boldsymbol{v}, \boldsymbol{b}) = u_i(0, v_{-i}, \boldsymbol{b}) + \int_0^{v_i} x_i(s, v_{-i}, \boldsymbol{b})\mathrm{d}s.$$

In other words, letting $p_i'(\boldsymbol{v}, \boldsymbol{b}) := p_i(\boldsymbol{v}, \boldsymbol{b}) - p_i(0, v_{-i}, \boldsymbol{b})$ and $u_i'(\boldsymbol{v}, \boldsymbol{b}) := v_i \cdot x_i(\boldsymbol{v}, \boldsymbol{b}) - p_i'(\boldsymbol{v}, \boldsymbol{b})$, we conclude that the utility of buyer i in this period is fully determined by the allocation function $x_i(\cdot, \boldsymbol{b})$ and the minimum payment $p_i(0, v_{-i}, \boldsymbol{b})$:

$$u_i'(\boldsymbol{v}, \boldsymbol{b}) = u_i(\boldsymbol{v}, \boldsymbol{b}) + p_i(0, v_{-i}, \boldsymbol{b}) = u_i(\boldsymbol{v}, \boldsymbol{b}) - u_i(0, v_{-i}, \boldsymbol{b}) + 0 \cdot x_i(0, v_{-i}, \boldsymbol{b}) = \textstyle\int_0^{v_i} x_i(s, v_{-i}, \boldsymbol{b})\mathrm{d}s.$$

Recall that for (BU), we require (i) the increase of balance cannot be more than the utility obtained in the current period and (ii) the updated balance must be nonnegative. Note that by the definition of $\xi_i(v_{-i})$ (1):

$$\mathbb{E}_{v_i}[u_i'(\boldsymbol{v}, \boldsymbol{b}) - p_i(0, v_{-i}, \boldsymbol{b})] = \mathbb{E}_{v_i}[u_i(\boldsymbol{v}, \boldsymbol{b})] = \xi_i(v_{-i}) \iff p_i(0, v_{-i}, \boldsymbol{b}) = \mathbb{E}_{v_i}[u_i'(\boldsymbol{v}, \boldsymbol{b})] - \xi_i(v_{-i}).$$

Hence on one hand, for (i) and (ii), we can rewrite it as follows:

$$\begin{aligned} & 0 \le b_i + \Delta b_i(\boldsymbol{v}) \le b_i + u_i(\boldsymbol{v}, \boldsymbol{b}) \\ \iff & -b_i \le \Delta b_i(\boldsymbol{v}) \le u_i(\boldsymbol{v}, \boldsymbol{b}) = u_i'(\boldsymbol{v}, \boldsymbol{b}) - p_i(0, v_{-i}, \boldsymbol{b}) = u_i'(\boldsymbol{v}, \boldsymbol{b}) - \mathbb{E}_{v_i}[u_i'(\boldsymbol{v}, \boldsymbol{b})] + \xi_i(v_{-i}). \end{aligned}$$

On the other hand, to ensure the set of the possible balance increment $\Delta b_i(\boldsymbol{v})$ satisfying (i) and (ii) is not empty, we need $b_i + u_i(\boldsymbol{v}, \boldsymbol{b}) \ge 0$. Note that b_i is given and $u_i(\boldsymbol{v}, \boldsymbol{b})$ reaches the minimum when $v_i = 0$, it is equivalent to:

$$b_i + u_i(0, v_{-i}, \boldsymbol{b}) = b_i - p_i(0, v_{-i}, \boldsymbol{b}) \ge 0 \iff \mathbb{E}_{v_i}[u_i'(\boldsymbol{v}, \boldsymbol{b})] \le b_i + \xi_i(v_{-i}).$$

Electronic copy available at: https://ssrn.com/abstract=3282600

In summary,

$$\text{(BU)} \iff \begin{cases} -b_i \le \Delta b_i(\boldsymbol{v}) \le u_i'(\boldsymbol{v},\boldsymbol{b}) - \mathbb{E}_{v_i}[u_i'(\boldsymbol{v},\boldsymbol{b})] + \xi_i(v_{-i}) \\ \mathbb{E}_{v_i}[u_i'(\boldsymbol{v},\boldsymbol{b})] \le b_i + \xi_i(v_{-i}) \end{cases}. \tag{4}$$

Finally, we also need the feasibility constraint of the allocation rule, that is, the total allocation of each item cannot be more than 1:

$$\textstyle\sum_i x_i(\boldsymbol{v},\boldsymbol{b}) \le 1. \tag{5}$$

Therefore, combining the objective (2) and the constraints (3), (4), and (5), for given $\boldsymbol{b}$ and g, the optimization program for current period can be written as:

$$\begin{aligned} \max \quad & \textsc{Rev}(\boldsymbol{b},\xi) = \mathbb{E}_{\boldsymbol{v}}\left[\textstyle\sum_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i})) + g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b}|M)\right] \\ \text{subj.t.} \quad & x_i(\boldsymbol{v}) \text{ is monotone in } v_i \text{ for any fixed } v_{-i} \\ & \mathbb{E}_{v_i}[u_i'(\boldsymbol{v})] \le b_i + \xi_i(v_{-i}),\ \forall i, v_{-i} \\ & -b_i \le \Delta b_i(\boldsymbol{v}) \le u_i'(\boldsymbol{v}) - \underset{v_i}{\mathbb{E}}[u_i'(\boldsymbol{v})] + \xi_i(v_{-i}),\ \forall i, \boldsymbol{v} \\ & \textstyle\sum_{i\in[n]} x_i(\boldsymbol{v}) \le 1,\ \forall \boldsymbol{v} \\ & x_i(\boldsymbol{v}) \ge 0, \Delta b_i(\boldsymbol{v}) \ge 0,\ \forall i, \boldsymbol{v} \end{aligned} \tag{6}$$

where we hide the $\boldsymbol{b}$ in the input to $x_i(\boldsymbol{v},\boldsymbol{b})$ and $u_i'(\boldsymbol{v},\boldsymbol{b})$ to emphasize that $\boldsymbol{b}$ is extraneous.

Simplification We then simplify the optimization program for optimal bank account mechanisms. One key observation is that with any fixed ξ (for all periods), the optimization with any finite horizon can be solved via backward induction.

In the program for the last period T, the expected revenue for the future, g^T, is always zero. For any parameter $\boldsymbol{b}$ of the last period, the optimal value of the program defines a function with respect to the balance vector $\boldsymbol{b}$, which, in fact, is the expected revenue for the last period where the mechanism in the last period M_T^* is optimal (for given $\boldsymbol{b}$ and ξ).

In other words, the optimal solution of the program for period T defines the function $g^{T-1}(\boldsymbol{b}|M_T^*)$, where the mechanism for period T is fixed to be optimal. Similarly, the optimal mechanism for each period M_t^* and the corresponding $g^t(\boldsymbol{b}|M_{t+1..T}^*)$ functions in the program for each period can be determined by backward induction. From now on, by omitting the mechanism being conditional on in g^t, we mean the g^t function is conditional on the optimal $M_{t+1..T}^*$, i.e.,

$$g^t(\boldsymbol{b}) := g^t(\boldsymbol{b}|M_{t+1..T}^*) = \max_M g^t(\boldsymbol{b}|M).$$

Note that the maximum always exists because the domain of the mechanisms is compact and the revenue is a continuous function of the mechanism.

Lemma 3.4. *$g^t(\boldsymbol{b})$ is weakly increasing.*

We place the proof of Lemma 3.4 in Section A.1. Hence we can eliminate variable $\Delta\boldsymbol{b}(\boldsymbol{v})$ from the original program (6): Since $g^t(\boldsymbol{b})$ is weakly increasing, it is without loss of generality to enforce

$$\Delta b_i(\boldsymbol{v}) = u_i'(\boldsymbol{v}) - \mathbb{E}_{v_i}[u_i'(\boldsymbol{v})] + \xi_i(v_{-i}),\ \forall i, \boldsymbol{v}.$$

Meanwhile, by the following Myerson's lemma, we can further rewrite $\mathbb{E}_{v_i}[u_i'(\boldsymbol{v})]$.

Electronic copy available at: https://ssrn.com/abstract=3282600

Lemma 3.5 (Myerson [1981]). *For any incentive compatible single item auction, the expected payment of buyer i equals to the expected (Myerson's) virtual welfare contribution of buyer i:*

$$\forall i, v_{-i},\ \ \mathbb{E}_{v_i}[p_i(\boldsymbol{v})] = \mathbb{E}_{v_i}[(v_i - \vartheta_i(\boldsymbol{v})) \cdot x_i(\boldsymbol{v})],$$

where $\vartheta_i(\boldsymbol{v}) = v_i \cdot (1 - F_i(v_i))/f_i(v_i)$ is the value divided by the hazard rate.[7]

Therefore,[8]

$$\mathbb{E}_{v_i}[u_i'(\boldsymbol{v})] = \mathbb{E}_{v_i}[v_i \cdot x_i(\boldsymbol{v}) - p_i'(\boldsymbol{v})] = \mathbb{E}_{v_i}[v_i \cdot x_i(\boldsymbol{v}) - (v_i - \vartheta_i(\boldsymbol{v})) \cdot x_i(\boldsymbol{v})] = \mathbb{E}_{v_i}[\vartheta_i(\boldsymbol{v}) \cdot x_i(\boldsymbol{v})]. \quad (7)$$

Thus, program (6) can be simplified as follows, where $\Delta b_i(\boldsymbol{v})$, $u_i'(\boldsymbol{v})$, and $\bar{u}_i'(v_{-i})$ are notations, not variables.

$$\begin{aligned} \max\quad & \textsc{Rev}(\boldsymbol{b}, \xi) = \mathbb{E}_{\boldsymbol{v}}\left[\sum\nolimits_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i})) + g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})\right] \quad (8)\\ \text{subj.t.}\quad & x_i(\boldsymbol{v}) \text{ is monotone in } v_i \text{ for fixed } v_{-i}\\ & \bar{u}_i'(v_{-i}) := \mathbb{E}_{v_i}[u_i'(\boldsymbol{v})] = \mathbb{E}_{v_i}[\vartheta_i(\boldsymbol{v}) x_i(\boldsymbol{v})] \le b_i + \xi_i(v_{-i}),\ \forall i, v_{-i}\\ & \Delta b_i(\boldsymbol{v}) = u_i'(\boldsymbol{v}) - \bar{u}_i'(v_{-i}) + \xi_i(v_{-i}),\ \forall i, \boldsymbol{v}\\ & \textstyle\sum_{i\in[n]} x_i(\boldsymbol{v}) \le 1,\ \forall \boldsymbol{v}\\ & x_i(\boldsymbol{v}) \ge 0,\ \forall i, \boldsymbol{v} \end{aligned}$$

Convexity We remains to show that the program is convex. In fact, we have the following lemma.

Lemma 3.6. *$g^t(\boldsymbol{b})$ is and concave.*

We place the proof of Lemma 3.6 in Section A.1. Since $g^t(\boldsymbol{b})$ is concave, the objective function (8) is also concave. Meanwhile, all the constraints are linear, so the program is a convex program (note that a standard convex program is minimizing a convex objective function or maximizing a concave objective function).

3.2 Duality, virtual values, and ironing

We first consider the optimal solution to the program with the monotonicity constraint (3) on $\boldsymbol{x}(\boldsymbol{v})$ relaxed. Later in Appendix B, we generalize the analysis to the original program. Although the optimal solution to the relaxed program is not a feasible solution, the analysis does capture the most interesting insight to this problem and provides the explicit form of the virtual values (before ironing). In fact, we show in Appendix B, that adding the monotonicity constraint (3) back to the program corresponds to applying the ironing operation on the virtual values.

For the following relaxed program, let $\lambda_i(v_{-i})$ be the Lagrange multiplier of the constraint $\mathbb{E}_{v_i}[\vartheta_i(\boldsymbol{v})x_i(\boldsymbol{v})] \le b_i + \xi_i(v_{-i})$ and $\mu(\boldsymbol{v})$ be the Lagrange multiplier of the feasibility constraint (5).

$$\begin{aligned} \max\quad & \textsc{Rev}(\boldsymbol{b}, \xi) = \mathbb{E}_{\boldsymbol{v}}\left[\sum\nolimits_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i})) + g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})\right] && \text{Lagrange multipliers} \quad (9)\\ \text{subj.t.}\quad & \bar{u}_i'(v_{-i}) := \mathbb{E}_{v_i}[u_i'(\boldsymbol{v})] = \mathbb{E}_{v_i}[\vartheta_i(\boldsymbol{v})x_i(\boldsymbol{v})] \le b_i + \xi_i(v_{-i}),\ \forall v_{-i} && \lambda_i(v_{-i})\\ & \Delta b_i(\boldsymbol{v}) = u_i'(\boldsymbol{v}) - \bar{u}_i'(v_{-i}) + \xi_i(v_{-i}),\ \forall i, \boldsymbol{v} && \varnothing\\ & \textstyle\sum_{i\in[n]} x_i(\boldsymbol{v}) \le 1,\ \forall \boldsymbol{v} && \mu(\boldsymbol{v})\\ & x_i(\boldsymbol{v}) \ge 0,\ \forall i, \boldsymbol{v} \end{aligned}$$

[7] F_i must be absolutely integrable.

[8] $\vartheta_i(\boldsymbol{v})$ is called *virtual value for utility* and equation (7) is from [Hartline and Roughgarden, 2008, Lemma 2.6].

Electronic copy available at: https://ssrn.com/abstract=3282600

Then the Lagrange $L := L(\boldsymbol{x}(\boldsymbol{v}), \lambda_i(v_{-i}), \mu(\boldsymbol{v}))$ is

$$\begin{aligned} L = &\mathbb{E}_{\boldsymbol{v}}\left[\textstyle\sum_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i})) + g(\Delta \boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})\right] \\ &- \textstyle\sum_{i, v_{-i}} \lambda_i(v_{-i}) \left(\mathbb{E}_{v_i}[\vartheta_i(\boldsymbol{v}) x_i(\boldsymbol{v})] - b_i - \xi_i(v_{-i})\right) - \sum_{\boldsymbol{v}} \mu(\boldsymbol{v}) \left(\textstyle\sum_i x_i(\boldsymbol{v}) - 1\right). \end{aligned}$$

Note that all constraints in program (9) are linear, therefore the Slater's condition is satisfied and the KKT conditions [Boyd and Vandenberghe, 2004, Chapter 5] are necessary and sufficient for any optimal solution to (9). In particular, the explicit form of the virtual values can be derived from the KKT conditions.

Let g_i denote the partial derivative of g with respect to the i-th dimension, i.e., $g_i(\boldsymbol{b}) = \partial g(\boldsymbol{b}) / \partial b_i$. Let α and β be defined as follows,

$$\alpha_i(\boldsymbol{v}) := 1 + g_i(\Delta \boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b}) \qquad \beta_i(v_{-i}) := \tfrac{\lambda_i(v_{-i})}{f(v_{-i})} + \mathbb{E}_{v_i}[g_i(\Delta \boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})].$$

Lemma 3.7 (Virtual welfare maximizer)**.** *Any optimal solution to* (9) *must maximize the expected virtual welfare, where the virtual value function* $\phi_i(\boldsymbol{v})$ *is given as follows:*

$$\phi_i(\boldsymbol{v}) = \alpha_i(\boldsymbol{v}) \cdot v_i - \beta_i(v_{-i}) \cdot \vartheta_i(\boldsymbol{v}).$$

We place the detailed proof of this lemma in Section A.2.

Ironing As we mentioned at the beginning of this subsection, the optimal solution to the original program (8) must maximize the expected ironed virtual welfare defined according to the ironed virtual values $\tilde{\phi}_i(\boldsymbol{v})$. Moreover, the ironed virtual value $\tilde{\phi}_i(\boldsymbol{v})$ is the transformation of $\phi_i(\boldsymbol{v})$ according to the ironing rules (Definition 3.1). We omit the details of the analysis here and incorporate the conclusion in the formal version of our first main theorem (Theorem 3.8). We refer the readers to Appendix B for the complete proof.

3.3 Sensitivity analysis and the optimality across the periods

So far we have determined the optimal auction for each period when the expected buyer utilities of each period ξ are fixed. Now we show how to optimize the remaining parameters ξ through sensitivity analysis. Let $R(\xi)$ denote the optimal revenue when the auctions in each period are optimal for the given ξ. Then we show how to determine the partial derivatives of $R(\xi)$ via sensitivity analysis and hence enable the gradient descent algorithm to find the optimal ξ^*. In particular, $R(\xi)$ is a concave function.[9]

The standard sensitivity analysis [Boyd and Vandenberghe, 2004, Chapter 5.6] indicates how much the optimal objective value of a program will change if some constraints become slightly looser or tighter. Such quantities usually have important physical meanings in economic setups. For example, consider $g^t(\boldsymbol{b}) = \max \textsc{Rev}^t(\boldsymbol{b}, \xi)$ (we brought back the superscript ^{t} to distinguish the variables for different periods). By sensitivity analysis,[10] we have:

$$g_i^t(\boldsymbol{b}) = \mathbb{E}_{\boldsymbol{v}^t}[g_i^{t+1}(\Delta \boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})] + \textstyle\sum_{v_{-i}^t} \lambda_i^t(v_{-i}^t).$$

Then $g_i^t(\boldsymbol{b})$, by definition, is the marginal contribution of the balance of buyer i in period t to the expected revenue since the t-th period. Similarly, $\mathbb{E}_{\boldsymbol{v}^t}[g_i^{t+1}(\Delta \boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})]$ is the marginal contribution

[9] Since by simply taking any convex combination of different ξ and ξ', the revenue obtained is also the convex combination of the revenue resulted by ξ and ξ'.

[10] If g^t is not differentiable at $\boldsymbol{b}$, the right-hand-side is a subgradient of g^t.

Electronic copy available at: https://ssrn.com/abstract=3282600

of b_i to the expected revenue since the $(t+1)$-th period. Hence their difference, $\sum_{v_{-i}^t} \lambda_i^t(v_{-i}^t)$, is the marginal contribution of b_i to the expected revenue of the t-th period. In particular, $\lambda_i^t(v_{-i}^t)$ is the marginal contribution if the values of other buyers are v_{-i}^t.

Moreover, we can conclude the concrete form of the partial derivatives of $R(\xi)$ by sensitivity analysis:

$$\begin{aligned} &\frac{\partial g^t(\boldsymbol{b})}{\partial \xi_i^t(v_{-i}^t)} = -f^t(v_{-i}^t) + f^t(v_{-i}^t) \cdot \mathbb{E}_{v_i^t}[g_i^{t+1}(\Delta \boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})] + \lambda_i^t(v_{-i}^t) = (\beta_i^t(v_{-i}^t) - 1) f^t(v_{-i}^t) \\ \Longrightarrow \quad &\frac{\partial R(\xi)}{\partial \xi_i^t(v_{-i}^t)} = \frac{\partial \mathbb{E}_{\boldsymbol{b}}[g^t(\boldsymbol{b})]}{\partial \xi_i^t(v_{-i}^t)} = (\mathbb{E}_{\boldsymbol{b}}\left[\beta_i^t(v_{-i}^t)\right] - 1) f^t(v_{-i}^t), \end{aligned}$$

where the expectation taken over $\boldsymbol{b}$ is computed by simulating the auctions in previous periods. In particular, ξ is selected optimally, if and only if:

$$\mathbb{E}_{\boldsymbol{b}}\left[\beta_i^t(v_{-i}^t)\right] = 1 \quad \text{or} \quad \xi^t(v_{-i}^t) = 0, \ \ \mathbb{E}_{\boldsymbol{b}}\left[\beta_i^t(v_{-i}^t)\right] \leq 1.$$

3.4 Summary

As a summary, we formally restate Theorem 3.2.

Theorem 3.8 (Formal)**.** *A bank account mechanism is optimal if and only if all the following conditions are satisfied:*

- *it satisfies all the basic constraints,* (IC)*,* (BI)*,* (BU) *and the feasibility constraint* (5)*;*
- *its allocation rule maximizes the ironed virtual welfare, where the virtual value (before ironing) $\phi(\boldsymbol{v})$ has the following form:*

  $$\phi(\boldsymbol{v}) = \alpha_i(\boldsymbol{v}) v_i - \beta_i(v_{-i}) \vartheta_i(\boldsymbol{v}),$$

  which can be seen as the combination of the Myerson's virtual value and the private value;
- *finally, the expected utility of each period ξ is selected optimally, i.e.,*

  $$\mathbb{E}_{\boldsymbol{b}}\left[\beta_i^t(v_{-i}^t)\right] = 1 \quad \textit{or} \quad \xi^t(v_{-i}^t) = 0, \ \ \mathbb{E}_{\boldsymbol{b}}\left[\beta_i^t(v_{-i}^t)\right] \leq 1.$$

4 An Algorithmic Approach to the Structure

So far we showed that the optimal bank account mechanism maximizes the ironed virtual welfare in each period. Although the explicit form of the virtual values (before ironing) is given in Lemma 3.7 and the ironing rule is given in Definition 3.1, how to accomplish the ironing operation is still unknown. One of the major difficulty of the ironing comes from the fact that one's virtual value not only depends on his/her own private value, but also depends on the private values of other buyers through the term $\alpha_i(\boldsymbol{v})$. In the presence of such interdependence across different buyers on virtual values, monotone virtual value function does not imply monotone allocation rules.

In this section, we algorithmically resolve the difficulty of ironing. In particular, we show a Fully Polynomial Time Approximation Scheme (FPTAS) that can accomplish the ironing step for any constant many buyer cases and hence compute the ironed virtual values. Moreover, the bank account mechanism induced by the ironed virtual values computed is (multiplicatively) $(1-\epsilon)$-approximately optimal in terms of revenue.

We also emphasize that the hard core of computing the exact optimal solution is not directly from the ironing step but the step of approximating the concave functions $g^t(\boldsymbol{b})$. Note that $g^t(\boldsymbol{b})$

Electronic copy available at: https://ssrn.com/abstract=3282600

is a continuous function without closed forms and the main effort of this section is to show that we can arbitrarily approximate $g^t(\boldsymbol{b})$ with piece-wise linear functions and guarantee that (i) the number of pieces is polynomial in the input size and (ii) the final result is approximately optimal.

Recall the program (12), in particular, we bring back the superscripts ^{t} to emphasize the periods:

$$\begin{aligned}
\max \quad & g^{t-1}(\boldsymbol{b}^t) = \textsc{Rev}(\boldsymbol{b}^t, \xi^t) = \mathbb{E}_{\boldsymbol{v}^t}\left[\sum\nolimits_i (v_i^t \cdot x_i^t(\boldsymbol{v}^t) - \xi_i^t(v_{-i}^t)) + g^t(\Delta \boldsymbol{b}^t(\boldsymbol{v}^t) + \boldsymbol{b}^t)\right] \\
\text{subj.t.} \quad & x_i^t(v_i^t, v_{-i}^t) - x_i^t(v_i^{t\prime}, v_{-i}^t) \le 0, \ \forall i, v_{-i}^t, v_i^{t\prime} > v_i^t \\
& \bar{u}_i^t(v_{-i}^t) := \mathbb{E}_{v_i^t}[u_i^{t\prime}] = \mathbb{E}_{v_i^t}[\vartheta_i^t(\boldsymbol{v}^t) x_i^t(\boldsymbol{v}^t)] \le b_i^t + \xi_i^t(v_{-i}^t), \ \forall v_{-i}^t \\
& \Delta b_i^t(\boldsymbol{v}^t) = u_i^{t\prime}(\boldsymbol{v}^t) - \bar{u}_i^t(v_{-i}^t) + \xi_i^t(v_{-i}^t), \ \forall i, \boldsymbol{v}^t \\
& \textstyle\sum_{i\in[n]} x_i^t(\boldsymbol{v}^t) \le 1, \ \forall \boldsymbol{v}^t \\
& x_i^t(\boldsymbol{v}^t) \ge 0, \ \forall i, \boldsymbol{v}^t
\end{aligned}$$

In what follows, we formalize the FPTAS via dynamic programming to compute the optimal ironed virtual values for the discrete type case.

Theorem 4.1. *The ironed virtual values of the optimal bank account mechanism can be computed through a dynamic programming based algorithm.*

Moreover, for any $\epsilon > 0$, there is an FPTAS to achieve an ϵ-approximation (multiplicative) of the optimal revenue.

We first outline the main idea of the algorithm: (i) for any fixed ξ, compute the ironed virtual values of the approximately optimal bank account mechanism; (ii) compute the optimal ξ using gradient descent. Since we have shown that the revenue of the bank account mechanism is a concave function with respect to ξ, the second step is standard and can be done with polynomially many queries to the (approximately) optimal revenue as a function of ξ. In what follows, we will focus on the first step.

To compute the ironed virtual value, we need to solve the dual program of (12), which, of course, is equivalent to solve the primal because the strong duality holds. Note that when ξ and $\boldsymbol{b}^t$ are fixed, (12) is a standard convex program with polynomially many linear constraints. Hence the optimal solution to its dual could be computed efficiently given oracle accesses to the concave function g^t. Then by the definition of g^{t-1}, the value of the optimal solution is $g^{t-1}(\boldsymbol{b}^t)$. Therefore, the concave function g^t can be evaluated recursively for each t and hence the ironed virtual values of the optimal bank account mechanism can be computed via standard dynamic programming as well.

However, the computation of the entire dynamic programming is not directly efficient: (i) if each g^t is computed recursively upon every query, the depth of the recursion could be up to T and hence the total number of queries required would be exponential in T; (ii) if each g^t is computed once for all possible $\boldsymbol{b}$ so that any further queries of g^t can be answered from precalculated values, then the number of input points where g^t need to be computed would be unbounded.

The key step to resolve these issues is to approximate the concave functions $g^t(\boldsymbol{b})$ by piece-wise linear functions each with at most polynomially many pieces. In addition, each of the piece-wise linear functions can be further expressed as the minimum of a set of affine functions. Hence both the original convex program (12) at each period t and its dual can be approximated by a polynomially large linear program.

The following lemma ensures that each g^t can be well approximated by a piece-wise linear function.

Electronic copy available at: https://ssrn.com/abstract=3282600

Lemma 4.2. *For all $t \in [T]$, g^t can be κ-approximated by two concave piece-wise linear functions, $\underline{g}^t$ and $\bar{g}^t$:*

$$\forall \boldsymbol{b},\ \underline{g}^t(\boldsymbol{b}) \le g^t(\boldsymbol{b}) \le \bar{g}^t(\boldsymbol{b}) \le \underline{g}^t(\boldsymbol{b}) + \kappa \max_{\boldsymbol{b}'} g^t(\boldsymbol{b}').$$

Moreover, each of $\underline{g}^t$ and $\bar{g}^t$ can be written as the minimum of at most polynomially many pieces.

With such approximations to the concave function g^t, we can approximately solve the convex program (12) by solving the following linear program, where each occurrence of g^t is replaced by $\bar{g}^t$:

$$\begin{aligned}
\max \quad & \bar{h}^{t-1}(\boldsymbol{b}^t) := \mathbb{E}_{\boldsymbol{v}^t}\left[\sum\nolimits_i (v_i^t \cdot x_i^t(\boldsymbol{v}^t) - \xi_i^t(v_{-i}^t)) + \bar{g}^t(\Delta \boldsymbol{b}^t(\boldsymbol{v}^t) + \boldsymbol{b}^t)\right] && (10)\\
\text{subj.t.} \quad & x_i^t(v_i^t, v_{-i}^t) - x_i^t(v_i^{t\prime}, v_{-i}^t) \le 0,\ \forall i, v_{-i}^t, v_i^{t\prime} > v_i^t \\
& \bar{u}_i^t(v_{-i}^t) := \mathbb{E}_{v_i^t}[u_i^{t\prime}] = \mathbb{E}_{v_i^t}[\vartheta_i^t(\boldsymbol{v}^t) x_i^t(\boldsymbol{v}^t)] \le b_i^t + \xi_i^t(v_{-i}^t),\ \forall v_{-i}^t \\
& \Delta b_i^t(\boldsymbol{v}^t) = u_i^{t\prime}(\boldsymbol{v}^t) - \bar{u}_i^t(v_{-i}^t) + \xi_i^t(v_{-i}^t),\ \forall i, \boldsymbol{v}^t \\
& \textstyle\sum_{i\in[n]} x_i^t(\boldsymbol{v}^t) \le 1,\ \forall \boldsymbol{v}^t \\
& x_i^t(\boldsymbol{v}^t) \ge 0,\ \forall i, \boldsymbol{v}^t \\
& \bar{g}^t(\Delta \boldsymbol{b}^t(\boldsymbol{v}^t) + \boldsymbol{b}^t) \le \boldsymbol{\alpha}_l \cdot (\Delta \boldsymbol{b}^t(\boldsymbol{v}^t) + \boldsymbol{b}^t) + \beta_l,\ \forall l, \boldsymbol{v}^t && (11)
\end{aligned}$$

As we mentioned previously, in the last constraint (11), we assume that the function $\bar{g}^t$ can be expressed by the minimum of a set of affine functions, i.e.,

$$\bar{g}^t(\boldsymbol{b}) = \min\nolimits_{l\in L} \boldsymbol{\alpha}_l \cdot \boldsymbol{b} + \beta_l.$$

Here we slightly abuse the notation of $\bar{g}^t(\Delta \boldsymbol{b}^t(\boldsymbol{v}^t) + \boldsymbol{b}^t)$ as variables in the linear program. Note that including the constraints (11) in the linear program (10) suffices to ensure that the variable values always agree with the corresponding function values. Because on the one hand, by the constraints, each variable is no more than the corresponding affine functions; on the other hand, since the coefficients of these variables are always positive in the objective, for each of the variables, at least one of the constraints in (11) must be binding.

Let $\bar{h}^{t-1}(\boldsymbol{b})$ denote the optimal value of the linear program (10) when $\boldsymbol{b}^t = \boldsymbol{b}$. Similarly, we can define $\underline{h}^{t-1}$ by simply replacing all $\bar{g}^t$ with $\underline{g}^t$ in (10). The following lemma shows that $\underline{h}^{t-1}$ and $\bar{h}^{t-1}$ are in fact lower and upper bounds of g^{t-1}.

Lemma 4.3. *$\underline{h}^{t-1}$ and $\bar{h}^{t-1}$ are concave and*

$$\forall \boldsymbol{b},\ \underline{h}^{t-1}(\boldsymbol{b}) \le g^{t-1}(\boldsymbol{b}) \le \bar{h}^{t-1}(\boldsymbol{b}) \le \underline{h}^{t-1}(\boldsymbol{b}) + \max_{\boldsymbol{b}'}(\bar{g}^t(\boldsymbol{b}') - \underline{g}^t(\boldsymbol{b}')).$$

Proof of Lemma 4.3. Let $\underline{\boldsymbol{x}}^*$, $\boldsymbol{x}^*$, and $\bar{\boldsymbol{x}}^*$ be the corresponding optimal solution for $\underline{h}^{t-1}(\boldsymbol{b})$, $g^{t-1}(\boldsymbol{b})$, and $\bar{h}^{t-1}(\boldsymbol{b})$, respectively. Denote these three programs as $\underline{P}$, P, and $\bar{P}$, respectively. Note that $\underline{\boldsymbol{x}}^*$ is feasible in P, hence by the optimality of $\boldsymbol{x}^*$,

$$P(\underline{\boldsymbol{x}}^*) \le P(\boldsymbol{x}^*),$$

where $P(\boldsymbol{x})$ denotes the objective value of program P with variables being $\boldsymbol{x}$.

On the other hand, for any $\boldsymbol{x}$ that is feasible to both $\underline{P}$ and P, we have,

$$\underline{P}(\boldsymbol{x}) \le P(\boldsymbol{x}).$$

Electronic copy available at: https://ssrn.com/abstract=3282600

Because in the objectives, $\underline{g}^{t-1} \le g^{t-1}$. Hence we conclude that

$$\underline{h}^{t-1}(\boldsymbol{b}) = \underline{P}(\underline{\boldsymbol{x}}^*) \le P(\underline{\boldsymbol{x}}^*) \le P(\boldsymbol{x}^*) = g^{t-1}(\boldsymbol{b}).$$

Similarly, we can prove that $g^{t-1}(\boldsymbol{b}) \le \bar{h}^{t-1}(\boldsymbol{b})$. For the last inequality, denote $\delta = \max_{\boldsymbol{b}'}(\bar{g}^t(\boldsymbol{b}') - \underline{g}^t(\boldsymbol{b}'))$. Note that for the $\bar{g}$ variables in $\bar{\boldsymbol{x}}^*$, if we reduce all of them by δ to get $\bar{\boldsymbol{x}}^{**}$, $\bar{\boldsymbol{x}}^{**}$ must be feasible to $\underline{P}$ and the objective value is reduced by at most δ, hence

$$\underline{P}(\boldsymbol{x}^*) \ge \underline{P}(\bar{\boldsymbol{x}}^{**}) \ge \bar{P}(\bar{\boldsymbol{x}}^*) - \delta.$$

In other words,

$$\underline{h}^{t-1}(\boldsymbol{b}) \ge \bar{h}^{t-1}(\boldsymbol{b}) - \delta.$$

□

Even through both $\underline{h}^{t-1}$ and $\bar{h}^{t-1}$ are in fact piece-wise linear functions, but they cannot be directly used for the computation of period $t-1$, because they may have exponentially many pieces. However, since they are concave, we can apply Lemma 4.2 to get the lower bound of $\underline{h}^{t-1}$ and the upper bound of $\bar{h}^{t-1}$:

$$\underline{g}^{t-1}(\boldsymbol{b}) \le \underline{h}^{t-1}(\boldsymbol{b}) \le g^{t-1}(\boldsymbol{b}) \le \bar{h}^{t-1}(\boldsymbol{b}) \le \bar{g}^{t-1}(\boldsymbol{b}).$$

Therefore, we can recursively compute $\underline{g}^1, \dots, \underline{g}^T$ and $\bar{g}^1, \dots, \bar{g}^T$ and hence compute the ironed virtual values of the approximately optimal bank account mechanism for any fixed ξ. Combining with the fact that ξ could then be optimized using gradient descent, we are done with our algorithm.

References

Itai Ashlagi, Constantinos Daskalakis, and Nima Haghpanah. Sequential mechanisms with ex-post participation guarantees. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 213–214. ACM, 2016. 2

Susan Athey and Ilya Segal. An efficient dynamic mechanism. *Econometrica*, 81(6):2463–2485, 2013. 2

Santiago Balseiro, Vahab Mirrokni, and Renato Paes Leme. Dynamic mechanisms with martingale utilities. 2016. 2

Santiago Balseiro, Max Lin, Vahab Mirrokni, Renato Paes Leme, and Song Zuo. Dynamic revenue sharing. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2678–2686, 2017. 2

Santiago R Balseiro, Vahab S Mirrokni, and Renato Paes Leme. Dynamic mechanisms with martingale utilities. forthcoming, *Management Science*, 2017. 2

Santiago Balseiro, Vahab Mirrokni, Renato Paes Leme, and Song Zuo. Dynamic double auctions: Towards first best. In *SODA 2019*. SIAM, 2019. 2

C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996. 18

Electronic copy available at: https://ssrn.com/abstract=3282600

Dirk Bergemann and Maher Said. Dynamic auctions. *Wiley Encyclopedia of Operations Research and Management Science*, 2011. 2

Dirk Bergemann and Juuso Välimäki. Information acquisition and efficient mechanism design. *Econometrica*, 70(3):1007–1033, 2002. 2

Dirk Bergemann and Juuso Välimäki. Dynamic mechanism design: An introduction. Technical report, Cowles Foundation for Research in Economics, Yale University, 2017. 2

Dirk Bergemann, Francisco Castro, and Gabriel Weintraub. The scope of sequential screening with ex post participation constraints. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 163–164. ACM, 2017. 2

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 10

Yang Cai and Mingfei Zhao. Simple mechanisms for subadditive buyers via duality. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 170–183, New York, NY, USA, 2017. ACM. 3

Yang Cai, Nikhil R Devanur, and S Matthew Weinberg. A duality-based unified approach to bayesian mechanism design. *ACM SIGecom Exchanges*, 15(1):71–77, 2016. 3

Ruggiero Cavallo. Efficiency and redistribution in dynamic mechanism design. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 220–229. ACM, 2008. 2

Shuchi Chawla, Nikhil R Devanur, Anna R Karlin, and Balasubranianian Sivan. Simple pricing schemes for consumers with evolving values. In *SODA 2016*, pages 1476–1490. SIAM, 2016. 2

Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. Mechanism design via optimal transport. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 269–286. ACM, 2013. 3

Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017. 3

Nikhil R Devanur, Yuval Peres, and Balasubramanian Sivan. Perfect bayesian equilibria in repeated sales. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 983–1002. SIAM, 2015. 2

Edith Elkind. Designing and learning optimal finite support auctions. In *SODA 2007*, pages 736–745. SIAM, 2007. 2

Jason D. Hartline and Tim Roughgarden. Optimal mechanism design and money burning. In *STOC 2008*, pages 75–84, 2008. 9

Matthew O Jackson and Hugo F Sonnenschein. Overcoming incentive constraints by linking decisions. *Econometrica*, 75(1):241–257, 2007. 2

Sham M Kakade, Ilan Lobel, and Hamid Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Operations Research*, 61(4):837–854, 2013. 2

Ilan Lobel and Renato Paes Leme. Dynamic mechanism design under positive commitment. 2017. 2

Electronic copy available at: https://ssrn.com/abstract=3282600

Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Dynamic auctions with bank accounts. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 387–393, 2016. 3, 5

Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Non-clairvoyant dynamic mechanism design. 2016. 5

Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Optimal dynamic mechanisms with ex-post ir via bank accounts. *arXiv preprint arXiv:1605.08840*, 2016. 3

Vahab Mirrokni, Renato Paes Leme, Rita Ren, and Song Zuo. Dynamic mechanism design in the field. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1359–1368. International World Wide Web Conferences Steering Committee, 2018. 3

Vahab Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Non-clairvoyant dynamic mechanism design. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 169–169. ACM, 2018. 2, 3

Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981. 2, 9

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007. 3

Mallesh M Pai and Rakesh Vohra. Optimal dynamic auctions and simple index rules. *Mathematics of Operations Research*, 38(4):682–697, 2013. 2

Christos Papadimitriou, George Pierrakos, Christos-Alexandros Psomas, and Aviad Rubinstein. On the complexity of dynamic mechanism design. In *SODA 2016*, pages 1458–1475. SIAM, 2016. 2

David C Parkes and Satinder Singh. An mdp-based approach to online mechanism design. 2004. 2

Alessandro Pavan, Ilya Segal, and Juuso Toikka. Dynamic mechanism design: A myersonian approach. *Econometrica*, 82(2):601–653, 2014. 2

Jean-Charles Rochet. The taxation principle and multi-time hamilton-jacobi equations. *Journal of Mathematical Economics*, 14(2):113–128, 1985. 7

Tim Roughgarden and Okke Schrijvers. Ironing in the dark. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 1–18. ACM, 2016. 2

Weiran Shen, Zihe Wang, and Song Zuo. Ex-post IR dynamic auctions with cost-per-action payments. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 505–511, 2018. 2

Electronic copy available at: https://ssrn.com/abstract=3282600

A Missing Proofs

A.1 Proof of Lemma 3.4 and Lemma 3.6

Proof of Lemma 3.4 and Lemma 3.6. We prove these lemmas by induction from $t = T$ down to 1. Since $g^T \equiv 0$, all these properties are satisfied for $t = T$.

Suppose that $g^{t+1}(\boldsymbol{b})$ is weakly increasing and concave. We first show that $g^t(\boldsymbol{b})$ is weakly increasing. Consider $\boldsymbol{b}' \geq \boldsymbol{b}$. By definition, $g^t(\boldsymbol{b})$ is the optimal value of program (6) for period t (with given ξ and $\boldsymbol{b}$). Note that:

1. The optimal solution for program (6) with balance $\boldsymbol{b}$ is also feasible for the program with balance $\boldsymbol{b}'$;
2. The objective is weakly increasing in $\boldsymbol{b}$ even with the same variables. The first part of the objective, $\mathbb{E}_{\boldsymbol{v}}[\sum_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i}))]$, is the same for the same allocation rule and the second part of the objective, $\mathbb{E}[g^{t+1}(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})]$, is weakly increasing in $\boldsymbol{b}$ by induction (with the same $\Delta\boldsymbol{b}(\boldsymbol{v})$).

Therefore, $g^t(\boldsymbol{b}') \geq g^t(\boldsymbol{b})$ is weakly increasing.

We then show that $g^t(\boldsymbol{b})$ is concave, i.e., for any $\theta \in [0, 1]$,

$$g^t(\theta\boldsymbol{b} + (1-\theta)\boldsymbol{b}') \geq \theta g^t(\boldsymbol{b}) + (1-\theta) g^t(\boldsymbol{b}').$$

Let $\boldsymbol{x}^*$ and $\Delta\boldsymbol{b}^*$ be the optimal solution for balance vector $\boldsymbol{b}$ and $\boldsymbol{x}^{**}$ and $\Delta\boldsymbol{b}^{**}$ be the optimal solution for balance vector $\boldsymbol{b}'$. Then their convex combination, $(\theta\boldsymbol{x}^* + (1-\theta)x^{**}, \theta\Delta\boldsymbol{b}^* + (1-\theta)\Delta\boldsymbol{b}^{**})$, is a feasible solution for balance vector $\theta\boldsymbol{b} + (1-\theta)\boldsymbol{b}'$. Meanwhile, the objective function is concave in the variables,

$$\begin{aligned}
& g^t(\theta\boldsymbol{b} + (1-\theta)\boldsymbol{b}') \\
\geq\ & \mathop{\mathbb{E}}_{v}\left[\sum_i (v_i \cdot (\theta x_i^*(\boldsymbol{v}) + (1-\theta)x_i^{**}(\boldsymbol{v})) - \xi_i(v_{-i})) + g^{t+1}(\theta(\Delta\boldsymbol{b}^*(\boldsymbol{v}) + \boldsymbol{b}) + (1-\theta)(\Delta\boldsymbol{b}^{**}(\boldsymbol{v}) + \boldsymbol{b}'))\right] \\
\geq\ & \theta \mathop{\mathbb{E}}_{v}\left[\sum_i (v_i \cdot x_i^*(\boldsymbol{v}) - \xi_i(v_{-i})) + g^{t+1}(\Delta\boldsymbol{b}^*(\boldsymbol{v}) + \boldsymbol{b})\right] \\
& + (1-\theta) \mathop{\mathbb{E}}_{v}\left[\sum_i (v_i \cdot x_i^{**}(\boldsymbol{v}) - \xi_i(v_{-i})) + g^{t+1}(\Delta\boldsymbol{b}^{**}(\boldsymbol{v}) + \boldsymbol{b}')\right] \\
=\ & \theta g^t(\boldsymbol{b}) + (1-\theta) g^t(\boldsymbol{b}'),
\end{aligned}$$

where the first inequality is by the definition of $g^t(\boldsymbol{b})$ and the feasibility of the convex combination we just argued and the second inequality is by the induction of the concavity of $g^{t+1}(\boldsymbol{b})$. □

A.2 Proof of Lemma 3.7

Proof of Lemma 3.7. By KKT conditions, $\partial L / \partial x_i(\boldsymbol{v}) \geq 0$:[11]

$$\begin{aligned}
\frac{\partial L}{\partial x_i(\boldsymbol{v})} &= \left(v_i - \frac{\lambda_i(v_{-i})\vartheta_i(\boldsymbol{v})}{f(v_{-i})}\right) f(\boldsymbol{v}) + \frac{\partial\, \mathbb{E}_{\boldsymbol{v}}[g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})]}{\partial x_i(\boldsymbol{v})} - \mu(\boldsymbol{v}) \\
&= \left(\left(1 + g_i(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})\right) v_i - \left(\frac{\lambda_i(v_{-i})}{f(v_{-i})} + \mathop{\mathbb{E}}_{v_i}[g_i(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})]\right)\vartheta_i(\boldsymbol{v})\right) f(\boldsymbol{v}) - \mu(\boldsymbol{v}) \\
&= (\alpha_i(\boldsymbol{v})v_i - \beta_i(v_{-i})\vartheta_i(\boldsymbol{v}))f(\boldsymbol{v}) - \mu(\boldsymbol{v}),
\end{aligned}$$

[11] If not differentiable, then any sub-gradient must be non-negative.

Electronic copy available at: https://ssrn.com/abstract=3282600

where

$$\begin{aligned}\frac{\partial \mathbb{E}_{\boldsymbol{v}}[g(\Delta \boldsymbol{b}(\boldsymbol{v})+\boldsymbol{b})]}{\partial x_i(\boldsymbol{v})} &= \frac{\partial \mathbb{E}_{\boldsymbol{v}}[g(\Delta \boldsymbol{b}(\boldsymbol{v})+\boldsymbol{b})]}{\partial u_i'(\boldsymbol{v})} \cdot \frac{\partial u_i'(\boldsymbol{v})}{\partial x_i(\boldsymbol{v})} + \frac{\partial \mathbb{E}_{\boldsymbol{v}}[g(\Delta \boldsymbol{b}(\boldsymbol{v})+\boldsymbol{b})]}{\partial \bar{u}_i'(v_{-i})} \cdot \frac{\partial \bar{u}_i'(v_{-i})}{\partial x_i(\boldsymbol{v})} \\ &= g_i(\Delta \boldsymbol{b}(\boldsymbol{v})+\boldsymbol{b}) f(\boldsymbol{v}) v_i - \mathop{\mathbb{E}}_{v_i}[g_i(\Delta \boldsymbol{b}(\boldsymbol{v})+\boldsymbol{b})] f(v_{-i}) \cdot f(v_i)\vartheta_i(\boldsymbol{v})\end{aligned}$$

and if $x_i(\boldsymbol{v}) > 0$, $\partial L / \partial x_i(\boldsymbol{v})$ must be 0, i.e.,

$$\begin{aligned}&\forall i,\ \boldsymbol{v},\ (\alpha_i(\boldsymbol{v}) v_i - \beta_i(v_{-i})\vartheta_i(\boldsymbol{v})) f(\boldsymbol{v}) - \mu(\boldsymbol{v}) \le 0 \\ &x_i(\boldsymbol{v}) > 0 \Longrightarrow (\alpha_i(\boldsymbol{v}) v_i - \beta_i(v_{-i})\vartheta_i(\boldsymbol{v})) f(\boldsymbol{v}) - \mu(\boldsymbol{v}) = 0\end{aligned}$$

Then for any $x_i(\boldsymbol{v}) > 0$, we have

$$\alpha_i(\boldsymbol{v}) v_i - \beta_i(v_{-i})\vartheta_i(\boldsymbol{v}) = \mu(\boldsymbol{v}) / f(\boldsymbol{v}) \ge \alpha_j(\boldsymbol{v}) v_j - \beta_j(v_{-j})\vartheta_j(\boldsymbol{v}).$$

In other words, the optimal solution is a virtual value maximizer, where the virtual value $\phi_i(\boldsymbol{v})$ is given as

$$\phi_i(\boldsymbol{v}) = \alpha_i(\boldsymbol{v}) v_i - \beta_i(v_{-i})\vartheta_i(\boldsymbol{v}).$$

In particular, $\lambda_i(v_{-i}) = 0$ when

$$\mathop{\mathbb{E}}_{v_i}[\vartheta_i(\boldsymbol{v}) x_i(\boldsymbol{v})] - b_i - \xi_i(v_{-i}) < 0,$$

i.e., the balance of buyer i is not spent out. □

A.3 Proof of Lemma 4.2

Proof of Lemma 4.2. The main idea of the proof is to (i) select polynomially many different points $\boldsymbol{b}$ in the k-dimension space $\mathbb{R}_+^k$, (ii) compute $g(\boldsymbol{b})$ for the selected points, and (iii) use the upper surface of the convex hull of $\{(\boldsymbol{b}, g(\boldsymbol{b}))\}$ as $\underline{h}(\boldsymbol{b})$.

Note that step (ii) and (iii) can be done very efficiently.[12] However, the step (i) is non-trivial.

To illustrate the algorithm, we first highlight that both the maximum value of each b_i, $g(\boldsymbol{b})$, and the partial derivatives $g_i(\boldsymbol{b})$ are bounded by $2^{O(N)}$, where N is the input size. In other words, simply selects points with fixed distance between other nearby points will result in exponentially many points. Therefore, the distances should be determined adaptively.

The key observation here is to keep track of an upper bound on the uncertainty of the partial derivative along each direction and the domain of b_i where the difference of $g(\boldsymbol{b})$ and $\underline{h}(\boldsymbol{b})$ could be more than $\kappa \cdot \max_{\boldsymbol{b}'} g(\boldsymbol{b}')$. Then each time pick a point on the boundary of the domain and the upper of the uncertainty of the partial derivative will be reduced by at least half. Hence after polynomially many steps, the uncertainty of the partial derivative will be exponentially small and we can easily select modest numbers of points to get a good enough approximation $\underline{h}(\boldsymbol{b})$ to the target function $g(\boldsymbol{b})$. □

[12] For (ii), the computation is efficient by hypothesis. For (iii), the convex hull in $(k+1)$-dimension can be using the quickhull algorithm [Barber *et al.*, 1996]. Note that k is a constant here.

Electronic copy available at: https://ssrn.com/abstract=3282600

B Ironing

We add the monotonicity constraint (3) back and show that this constraint corresponds to applying ironing operation on the virtual value functions.

Consider the following program without relaxation, where the monotonicity constraint is rewritten in an explicit way.

$$\begin{aligned}
\max \quad & \textsc{Rev}(\boldsymbol{b},\xi) = \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[\sum\nolimits_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i})) + g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})\right] && \text{Lagrange multipliers} \\
\text{subj.t.} \quad & x_i(v_i, v_{-i}) - x_i(v_i', v_{-i}) \le 0,\ \forall i, v_{-i}, v_i' > v_i && \eta_i(v_i, v_i'; v_{-i}) \\
& \bar{u}_i'(v_{-i}) \le b_i + \xi_i(v_{-i}),\ \forall i, v_{-i} && \lambda_i(v_{-i}) \\
& \Delta b_i(\boldsymbol{v}) = u_i'(\boldsymbol{v}) - \bar{u}_i'(v_{-i}) + \xi_i(v_{-i}),\ \forall i, \boldsymbol{v} && \varnothing \\
& \textstyle\sum_{i\in[n]} x_i(\boldsymbol{v}) \le 1,\ \forall \boldsymbol{v} && \mu(\boldsymbol{v}) \\
& x_i(\boldsymbol{v}) \ge 0,\ \forall i, \boldsymbol{v}
\end{aligned} \tag{12}$$

Then the Lagrange for this program is as follows.

$$\begin{aligned}
& L(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\
=& \mathop{\mathbb{E}}_{\boldsymbol{v}}\left[\sum_i (v_i \cdot x_i(\boldsymbol{v}) - \xi_i(v_{-i})) + g(\Delta\boldsymbol{b}(\boldsymbol{v}) + \boldsymbol{b})\right] \\
& - \sum_i \sum_{\boldsymbol{v}} x_i(\boldsymbol{v}) \left(\sum_{v_i' > v_i} \eta_i(v_i, v_i'; v_{-i}) - \sum_{v_i'' < v_i} \eta_i(v_i'', v_i; v_{-i})\right) \\
& - \sum_{i, v_{-i}} \lambda_i(v_{-i}) \left(\mathop{\mathbb{E}}_{v_i}[\vartheta_i(\boldsymbol{v}) x_i(\boldsymbol{v})] - b_i - \xi_i(v_{-i})\right) - \sum_{\boldsymbol{v}} \mu(\boldsymbol{v}) \left(\sum_i x_i(\boldsymbol{v}) - 1\right).
\end{aligned} \tag{L}$$

By similar argument with the proof of Lemma 3.7, any optimal auction must maximize the ironed virtual welfare and the *ironed virtual value function* $\tilde{\phi}_i(\boldsymbol{v})$ is given as

$$\tilde{\phi}_i(\boldsymbol{v}) = \alpha_i(\boldsymbol{v}) v_i - \beta_i(v_{-i}) \vartheta_i(\boldsymbol{v}) - \frac{1}{f(\boldsymbol{v})} \left(\sum_{v_i' > v_i} \eta_i(v_i, v_i'; v_{-i}) - \sum_{v_i'' < v_i} \eta_i(v_i'', v_i; v_{-i})\right).$$

Note that conditional on any v_{-i}, the expectation of the virtual value with ironing equals to the expectation of the virtual value without ironing:

$$\sum_{v_i} f_i(v_i) \phi_i(\boldsymbol{v}) = \sum_{v_i} f_i(v_i) \tilde{\phi}_i(\boldsymbol{v}).$$

Thus $\eta_i(v_i, v_i'; v_{-i})/f(v_{-i})$ defines a mass move of $f_i(v_i)\phi_i(\boldsymbol{v})$ within $\mathbb{R}_+$. In particular, since $\eta_i(v_i, v_i'; v_{-i}) \ge 0$, the move is from small v_i to large v_i' ($v_i < v_i'$).

Meanwhile, by complementary slackness,

$$(x_i(v_i, v_{-i}) - x_i(v_i', v_{-i}))\eta_i(v_i, v_i'; v_{-i}) = 0,$$

which means the mass move must happen within the regions with the same allocation.

Also note that the virtual value $\phi_i(\boldsymbol{v})$ depend on v_{-i} as well, the ironing does not make $\tilde{\phi}_i(\boldsymbol{v})$ an increasing function. Instead, the following function must be weakly increasing,

$$\operatorname{sgn}\left[\tilde{\phi}_i(\boldsymbol{v}) - \max\left\{0, \max_{j\neq i} \tilde{\phi}_j(\boldsymbol{v})\right\}\right].$$

Because the allocation maximizes the virtual welfare and must be weakly increasing.

Electronic copy available at: https://ssrn.com/abstract=3282600