Efficient Grid Search to Solve Static Games with Private Information*

Jakub Kastl† Eric Richert‡ Jesper Rüdiger§

April 2026

Abstract

We introduce a new method for computing equilibria in a large class of games, including auctions, nonlinear pricing and optimal contracting. First, we observe that a function, for instance the bid strategy function for bidders in an auction, can often be approximated arbitrarily well by piecewise constant functions on a discrete grid. Second, we formulate a sequential program to find the *global solution* within this class of strategies. This presents a major advantage over other solution methods which rely on local optimization. A simulation study of asymmetric auctions and nonlinear pricing games suggests that the method is stable, fast, and easily extends to complex games. We then examine two applications to highway procurement auctions and show that our method has higher accuracy and speed. Finally, we use our method to analyze menu design. We study the breakfast cereal market and compute counterfactual package size and price adjustments following a cost shock and a merger.

Keywords: equilibrium computation, auctions, nonlinear pricing, simulation, structural estimation

*We thank Paulo Somaini (our discussant) for helpful feedback, and Sofia Moroni and many seminar participants for useful comments and Renjie Bao for research assistance. All remaining errors are our own. Researcher(s)' own analyses calculated (or derived) based in part on data from Nielsen Consumer LLC and marketing databases provided through the NielsenIQ Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. The conclusions drawn from the NielsenIQ data are those of the researcher(s) and do not reflect the views of NielsenIQ. NielsenIQ is not responsible for, had no role in, and was not involved in analyzing and preparing the results reported herein.

†Princeton University, Department of Economics, NBER and CEPR

‡University of Chicago, Department of Economics, NBER

§Universidad Carlos III de Madrid, Department of Business Administration

1 Introduction

In many games, such as auctions, nonlinear pricing, optimal contracts and menu design, the equilibrium is a functional fixed point that is hard to compute. Yet we often want to solve the game, for example, to make counterfactual predictions on the effect of subsidies, mergers, technology changes or adjustments to market design.

A common approach is to use numerical methods to approximate the solution. A simple method would be to restrict the game's strategy space to a discrete grid, and then do an exhaustive grid search of all combinations to find the global solution. However, for even a small number of grid points, this becomes computationally intractable. Therefore, standard solution methods either rely on restrictive parametric forms or on path-dependence from a differential equation representation of the problem. Although these methods offer computational savings relative to a naïve grid search, they produce local solutions that are often unstable and sensitive to initial values. Furthermore, they are usually inflexible and must be adapted for each new setting, and even so, can still be slow to compute.

In this paper, we propose a new method for solving for equilibrium. We rely on two observations. First, in many games the players' best-response functions, such as a tariff function in price discrimination games or a bid function in auctions, can be arbitrarily closely approximated by restricting the strategy space to piece-wise constant step functions.[1] This restriction transforms the original environment into a *discretized game*. We show that the equilibria of the discretized game converge to those of the original game as the step size vanishes. Second, in a large class of games with 'local constraints', the players' optimization problem in the discretized game can be characterized by a sequential program. This program can be solved via a *recursive algorithm* which finds the global solution in the set of step functions.

We analyze the performance of the algorithm in two ways. First, we conduct a simulation study to benchmark against existing methods. We show that our recursive algorithm (a) accurately approximates profits; (b) is fast; (c) works over a wide range of examples without any need to adapt the code and extends easily to complex settings. Second, we re-examine two studies of highway procurement auctions that rely on existing methods, and show that our recursive algorithm increases accuracy, and leads to computational gains. Finally, we provide an application to the market for ready-to-eat breakfast cereal. We solve a model of the joint decision of box-sizes and prices and evaluate the impact of a marginal cost shock and a hypothetical merger

[1] We show that the equilibrium profits converge at a quadratic rate in the number of steps.

on prices and package sizes. Our method allows us to solve the simultaneous menu design of 25 product lines by 4 firms with an endogenous number of price-quantity pairs per product.

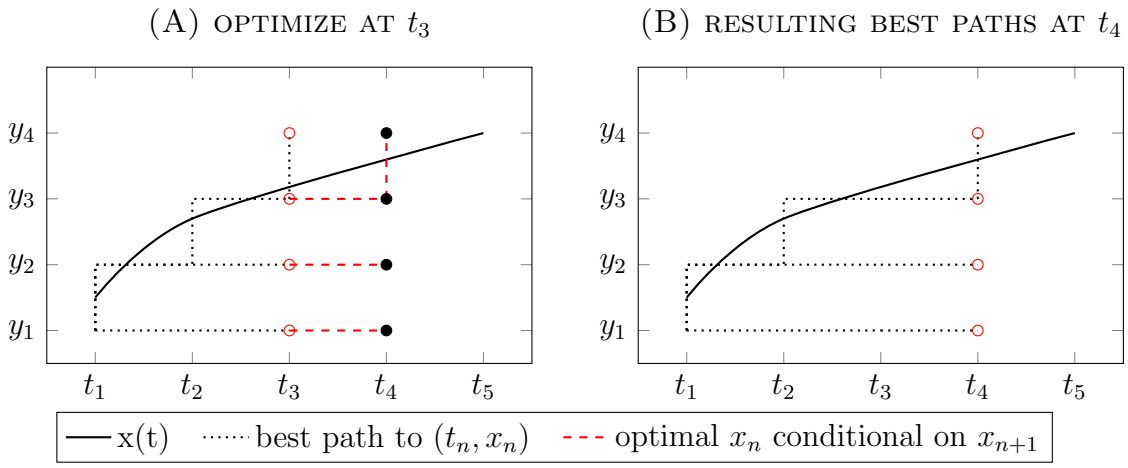
The optimization over step functions can be performed quickly and reliably because the problem can be decomposed into a number of independent sub-problems, allowing it to be solved using a recursive algorithm that leverages dynamic programming techniques. We show that this approach is valid when the problem has *local constraints*: the feasibility of a given strategy profile depends on a set of nearest-neighbor constraints. In particular, if $(t_1, \ldots, t_N)$ is a type space, and $x = (x_1, \ldots, x_N)$ a strategy function assigning an action from a discrete grid to each type, then the problem admits a local step structure if the feasibility of x can be checked via a set of constraints $\rho_n(x_n, x_{n+1}) \leq 0$, $n = 1, \ldots, N-1$. This holds for instance in auctions and in settings where local incentive compatibility implies global incentive compatibility, as in Carroll (2012).

To illustrate the method, Figure 1 shows how to solve for a bidder's optimal inverse strategy, $x(t)$, in a single-unit auction. Our objective is to approximate $x(t)$, represented by the solid line, over a discrete grid of valuations, $y_1, ..., y_4$, and prices, $t_1, ..., t_5$. At each price step n we choose the lowest valuation x_n that bids at price t_n, subject to local monotonicity constraints, $x_n \leq x_{n+1}$. Hence, the action at step n is constrained only by the action at step $n+1$. This structure allows for a sequential solution. At the first price step, for each potential x_2 we calculate our optimal choice of x_1. At the n'th price step, for each potential x_{n+1} we calculate our optimal choice of x_n. The key observation is that the optimal choice of x_{n-1} conditional on x_n remains invariant to the value of x_{n+1}. Consequently, at any step n, we can fix the optimal path up to each potential x_n, and compare only the set of feasible x_n values conditional on x_{n+1}.

To reinforce the recursive logic, panel A of Figure 1 illustrates the third price step. Filled circles represent potential x_4 values and empty circles potential x_3 values. Dotted lines represent the "best paths" leading up to each of the potential x_3 (i.e. optimal choices of x_1 and x_2 given x_3). Dashed lines represent the optimal x_3 for each possible x_4. For instance, if $x_4 = y_2$, then $x_3 = y_2$ is optimal, whereas if $x_4 = y_4$, then $x_3 = y_3$ is optimal. Panel B displays the resulting best paths leading up to each potential x_4. Note that while a path leading to $x_3 = y_4$ exists in panel A, it disappears in panel B as it is eliminated by the optimal choices at the previous step.

One way to think about the recursive algorithm is an "intelligent grid search", where we continually eliminate paths that cannot be part of the global solution. This

FIGURE 1: OPTIMIZATION WITH RECURSIVE ALGORITHM



approach finds the global solution as would a full grid search, but allows us to move much faster through the grid. This allows us to find a global solution very efficiently.[2]

The previous arguments showed how to calculate a player's best response, and we call this the *recursive single-agent algorithm*. To compute the equilibrium, where opponents' strategies can also vary, we extend the single-agent algorithm in two different ways. First, the *recursive full equilibrium algorithm* extends the state variables in the optimization procedure described above to include each opponent's behavior and enforces mutual best-response conditions at each step along the path. We show that if the local constraints assumption holds jointly for all players, then this approach will find the (global) equilibrium of the discretized game. However, the dimension of the state space of the dynamic programming problem grows at a rate equal to the number of players. Second, the *recursive iterated equilibrium algorithm* only requires the local constraints assumption to hold individually for each player. It avoids increasing the dimension with the number of bidders by iterating such that each iteration fixes opponent behavior and solves the individual problem using the single-agent recursive algorithm, then updates opponent behavior in search of a fixed point. Although this approach is not guaranteed to converge, it performs well in our simulations.

We next describe the results of our simulations. First, we apply the method to compute the equilibrium bid functions in a set of asymmetric auctions. We begin with examples where the bid function can be solved analytically or using standard numerical tools. In terms of speed, our recursive iterated equilibrium algorithm outperforms

[2]We show in Section 3 that if the price and action grids have N and K steps, respectively, then the number of calculations required to find a player's optimal strategy is $N \times K^2$ in our recursive algorithm but K^N in a full grid search.

the shooting algorithm and is comparable to the polynomial collocation approach of Hubbard and Paarsch (2014). In terms of accuracy, the polynomial collocation method is shown to be much more sensitive to initial values than our method, which is stable for a wide range of parameter values. We then consider some extensions that cannot be easily solved using the existing numerical methods such as an application with multiple types of bidders who draw values from distributions with non-common supports. Our method has comparable accuracy and similar or better solution times than the specialized problem-specific algorithms of Hubbard and Kirkegaard (2019) and Dharanan and Ellis (2024). The recursive full equilibrium algorithm has stable and accurate results in the examples but with higher compute times. Notice that this algorithm does not require starting values, and that it is therefore oblivious to this type of instability.

Second, we apply the method to solve for the equilibrium in the nonlinear pricing game of Martimort and Stole (2009). Our approach allows the researcher to include the full set of local constraints, rather than the relaxed program that is typically used and which assumes only the downward local incentive compatibility constraints are binding. In this setting, the incentive constraints add additional complexity and nonlinearity to the problem, and the recursive iterated equilibrium algorithm performs much better than polynomial approaches, both in terms of accuracy and speed. We also illustrate the performance of the method in a setting where the monopolist is choosing contracts for multidimensional types, and show how iteration of the recursive algorithm can be used to reduce the dimension of the state space.

We then examine two applications to highway procurement auctions. In the first, Krasnokutskaya and Seim (2011) use a shooting algorithm to analyze the California Small Business Preference program which gives preference to small firms in procurement auctions, thus creating (further) bidder asymmetry. We highlight two features: first, in settings with more than two asymmetric bidders, bid separation is possible (Hubbard and Kirkegaard, 2019), but this is ruled out by the boundary condition imposed in most empirical studies, including Krasnokutskaya and Seim (2011); second, the shooting algorithm can be unstable, and we illustrate patterns in their solution similar to those predicted by Fibich and Gavish (2011). Using their data and model estimates we analyze an auction where our method computes an equilibrium that allows for bid separation, and which is not subject to the shooting method instabilities.

In the second application, Somaini (2020) uses a shooting algorithm to analyze Michigan highway procurement auctions that feature interdependent costs. The interdependent costs introduce a winner's curse, potentially making it more difficult

to solve for equilibrium. We first benchmark his shooting algorithm by calculating the equilibrium of an auction with independent, uniformly distributed private values for different numbers of bidders. Consistent with the results of our simulations, our method is more accurate and faster than the shooting solution. We then explore the performance under alternative cost functions with interdependence, and again illustrate an improvement in convergence.

Finally, we apply the model to study the joint determination of box-sizes and prices of ready-to-eat breakfast cereals. Motivated by recent concerns of shrinkflation, we model the impact of a marginal cost shock on prices and package sizes. Menu adjustments are difficult to incorporate using existing tools due to their combinatorial nature, and previous studies usually restrict the product variety firms can offer for tractability (Fan and Yang, 2020; Wollmann, 2018; Eizenberg, 2014; Draganska, Mazzeo, and Seim, 2009). To explore the importance of endogenous box size adjustments, we consider a cost shock and compare a simple model in which firms hold fixed box-sizes and re-optimize prices to a full model where firms can adjust both sizes and prices. In the simple model the cost shock results in almost double (7.5%) the loss in consumer surplus as in the full model where firms also adjust their package sizes (5.4%). In addition, the simple model predicts a different pattern in the changes in profits earned across firms than the full model. We then apply the method to study the change in box-sizes following a merger, and again find that allowing for box-size adjustment has a meaningful impact on welfare, and counterfactual profit predictions. This suggests that ignoring box-sizes adjustments understates the level of market competition and firms' ability to respond to shocks.

The combined results of the paper suggest that our recursive algorithm can be applied in a broad range of settings, and that it offers advantages in terms of stability and speed over existing methods. These advantages can be leveraged to conduct counterfactual experiments and to model complex settings.[3] We show this in the context of auctions, nonlinear pricing, and menu choice, but the generality of the recursive algorithm and the ease with which it can be adapted to different settings suggest that it can be applied to many more topics such as complex subsidy design at auctions, optimal principal-agent contracts, and market entry and exit. Furthermore, the recursive full equilibrium algorithm can be extended to capture equilibrium multiplicity and can in principle find the global set of equilibria. In conclusion, we imagine that the recursive method can be useful in a broad range of scenarios.

[3]For instance, Gonzalez-Eiras, Kastl, and Rüdiger (2023) applies a single-agent version of our method to study dynamic incentives in Treasury bill auctions.

The rest of the paper is organized as follows. Section 2 introduces the class of games, to which our method applies. Section 3 describes the computational method in detail. Section 4 displays the results of our simulations. Section 5 contains the application to highway procurement auctions. Section 6 contains the application to the ready-to-eat breakfast cereal market. Finally, Section 7 concludes.

2 Games

This section introduces the class of static games with private information for which the recursive method applies. The key features are: (i) players choose functions (e.g. bid or tariff schedules) from an ordered type space into a finite or compact action set; (ii) payoffs can be written as integrals of a bounded, measurable payoff kernel over types and signals; and (iii) constraints on players' strategies are local. These properties guarantee that we can approximate the equilibria of the original game with equilibria of a discretized game, in a sense developed in later sections, and leverage the recursive structure.

2.1 Environment

There is a finite set of players $\mathcal{N} = \{1, \ldots, I\}$. Each player i draws a privately observed type θ_i from a compact, totally ordered type space $T_i \subset \mathbb{R}$ with Borel σ–algebra $\mathcal{B}_i$. Types are drawn from a commonly known joint distribution F on $T = \prod_{i \in \mathcal{N}} T_i$, with density f with respect to a product measure. We write $\theta = (\theta_i)_{i \in \mathcal{N}}$, with F_i and f_i for the marginal distributions of player i.

Each player i chooses a (possibly vector–valued) action in a compact metric action space $Y_i \subset \mathbb{R}^{k_i}$, where k_i is the dimensionality of the action. A (pure) strategy of player i is a measurable function

$$x_i : T_i \to Y_i.$$

We denote the profile by $x = (x_i)_{i \in \mathcal{N}}$, and write x_{-i} for the profile of opponents' strategies. The induced random action profile is $x(\theta) = (x_i(\theta_i))_{i \in \mathcal{N}}$.

2.2 Payoffs

For each player i, the ex–post payoff from action profile $y = (y_j)_{j\in\mathcal{N}}$, with $y \in Y \equiv \prod_{j\in\mathcal{N}} Y_j$, and type profile θ is given by a bounded, measurable function

$$\pi_i : Y \times T \to \mathbb{R}.$$

Given a strategy profile x, player i's expected payoff is

$$U_i(x) \;\equiv\; \int_T \pi_i\big(x(\theta),\theta\big)\, dF(\theta).$$

In many applications, π_i is generated by an auction or a nonlinear pricing game. For example, in a first–price auction with independent private values, T_i is the support of player i's value, Y_i is a compact interval of bids, $x_i(\theta_i)$ is a bid function, and π_i is the product of the probability of winning at that bid and the surplus conditional on winning. In nonlinear pricing, T_i is the type space, Y_i is the set of price–quantity pairs offered to type θ_i, and π_i is the profit from the chosen contract. These examples fit into the general form above.

2.3 Constraints and equilibrium

Players may face constraints on their strategy functions, such as monotonicity or incentive compatibility. The set of constraints is characterized by a continuous function R_i such that $x_i \in X_i$ if and only if for almost all $\theta \le \theta'$ in T_i:[4]

$$R_i(x_i(\theta),\theta; x_i(\theta'),\theta') \le 0. \tag{1}$$

For each i, let X_i denote the set of measurable, admissible strategies $x_i : T_i \to Y_i$ that satisfy these constraints. Let $X = \prod_{i\in\mathcal{N}} X_i$ be the set of admissible profiles. We refer to the game characterized above as the *unrestricted game*.

A pure–strategy Bayes–Nash equilibrium is a profile $x^* \in X$ such that for each i,

$$U_i(x^*) \;\ge\; U_i(x_i, x^*_{-i}) \quad \text{for all } x_i \in X_i.$$

Let X^* denote the set of equilibria of the game.

[4]For simplicity, constraints depend only on own actions, but other players' actions could be incorporated as well.

2.4 Discretized games

To compute equilibria numerically, we approximate the unrestricted game by a finite game. We propose approximating x_i by step functions and constructing best responses by solving a sequential program. This section demonstrates that the equilibria of these discretized games converge to the equilibria of the original game and lays out the assumption needed for the solution algorithm to work; the algorithm itself is detailed in Section 3.

2.4.1 Grid construction and strategy space

We partition each player i's type space into N_i^s non-overlapping subsets $(T_{i,n}^s)_{n=1}^{N_i^s}$, where the integer s is an index. From this partitioning, we construct a finite grid $\mathcal{T}_i^s \equiv \{t_{i,1}, \ldots, t_{i,N_i^s}\}$ with $t_{i,n}$ being the midpoint of $T_{i,n}^s$ for all i and n. Similarly, partition Y_i into K_i^s non-overlapping subsets $(Y_{i,k}^s)_{k=1}^{K_i^s}$, and construct a finite action grid $\mathcal{Y}_i^s \equiv \{y_{i,1}, \ldots, y_{i,K_i^s}\}$ where $y_{i,k}$ is the midpoint of $Y_{i,k}^s$ for all i and k.[5]

To construct a step function on these grids that satisfy the constraints of the unrestricted game, we define the projection operators $Q_i^s : T_i \to \mathcal{T}_i^s$ such that $Q_i^s(\theta) = t_{i,n}$ whenever $\theta \in T_{i,n}^s$ and $P_i^s : Y_i \to \mathcal{Y}_i^s$ such that $P_i^s(y) = y_{i,k}$ whenever $y \in Y_{i,k}^s$. The admissible set of step-function strategies is then the projection of the original set of admissible strategies onto the grids:

$$X_i^s \equiv \{\tilde{x}_i \in P_i^s \circ X_i \circ Q_i^s \mid R_i(\tilde{x}_i(t), t; \tilde{x}_i(t'), t') \leq 0 \text{ for all } t, t' \in \mathcal{T}_i^s\}.$$

Let $X^s \equiv \prod_{i \in \mathcal{N}} X_i^s$ be the set of admissible profiles.

A *discretized game* is defined by the admissible strategies X^s and payoffs for $x^s \in X^s$:

$$U_i(x^s) = \sum_{t \in \mathcal{T}^s} \int_{\theta \in T^s(t)} \pi_i(x^s(\theta), \theta) f(\theta)\, d\theta, \tag{2}$$

where $T^s(t)$ is the unique joint type partition containing t. Similar discretization approaches have been used to prove existence of auction equilibria (e.g., Reny and Zamir, 2004; Kastl, 2012).

2.4.2 Assumptions

We now describe two assumptions. The first assumption establishes the local constraints structure used in the recursive algorithm developed in Section 3. The second

[5] We construct the grids using midpoints for simplicity, but any interval point could be used. Note also that the partitioning need not consist of cells of equal size.

assumption establishes sufficient conditions for the discretized games to converge.

Assumption 1. *For each i, the constraints are* local*: for any discretization $(T^s_{i,n})_{n=1}^{N^s_i}$ and $(Y^s_{i,k})_{k=1}^{K^s_i}$, there exist functions $\rho^s_{i,n}$ such that a step function $\tilde{x}^s_i$ belongs to X^s_i if and only if its values on the grid, $\hat{x}_{i,n} \equiv \tilde{x}^s_i(t_{i,n})$, satisfy*

$$\rho^s_{i,n}\big(\hat{x}_{i,n}, \hat{x}_{i,n+1}\big) \;\leq\; 0 \quad \textit{for all } n = 1, \ldots, N^s_i - 1,$$

together with possibly additional one–point constraints at the boundary (e.g. at $t_{i,1}$).

The assumption ensures that the global constraints R_i can be characterized by a sequence of local constraints. Assumption 1 covers many standard constraints in mechanism design and auction theory, including monotonicity and settings where local incentive compatibility implies global incentive compatibility (e.g. Carroll, 2012).

Assumption 2. *The unrestricted game G satisfies the following:*

(i). For each player i, Y_i is a lattice, and $\pi_i(y, \theta)$ is bounded, continuous, weakly quasisupermodular and satisfies the weak single-crossing property.[6]

(ii). The unrestricted game G is better-reply secure.[7]

The assumption guarantees the existence of pure-strategy equilibria in both the unrestricted and discretized games by Reny (2011). While standard auctions feature discontinuities, this can be overcome by choosing disjoint action partitions to avoid ties, as in Reny and Zamir (2004).

2.4.3 Convergence results

We are now ready to state this section's main result, which describes the convergence of discretized equilibria such as those calculated algorithmically in the next section.

Theorem 1 (Convergence of discretized equilibria)**.** *Suppose Assumption 2 is satisfied. Let $(T^s_{i,n})_{n=1}^{N^s_i}$ and $(Y^s_{i,k})_{k=1}^{K^s_i}$ be such that as $s \to \infty$,*

$$\Delta^T_s \;\equiv\; \max_i \max_n \operatorname{diam}(T^s_{i,n}) \to 0, \qquad \Delta^Y_s \;\equiv\; \max_i \max_k \operatorname{diam}(Y^s_{i,k}) \to 0.$$

[6] Weak single-crossing requires that if $x^H_i \geq x^L_i$ and $\theta^H \geq \theta^L$, then for all monotone strategies x_{-i} of other players, $\pi_i((x^H_i, x_{-i}), \theta^L) \geq \pi_i((x^L_i, x_{-i}), \theta^L)$ implies $\pi_i((x^H_i, x_{-i}), \theta^H) \geq \pi_i((x^L_i, x_{-i}), \theta^H)$.

[7] G is *better-reply secure* if for every non-equilibrium strategy profile x and payoff vector u such that (x, u) is in the closure of the graph of $(U_i)_{i=1}^{\mathcal{N}}$, there exists a player i, a deviation $x'_i \in X_i$, a neighborhood V of x_{-i}, and an $\epsilon > 0$ such that $U_i(x'_i, y_{-i}) > u_i + \epsilon$ for all $y_{-i} \in V$.

Let G^s be the corresponding discretized game and $X^{s,*}$ its set of pure-strategy Bayes-Nash equilibria. Consider a sequence $\{x^{s,*}\}_{s\geq 1}$ where $x^{s,*} \in X^{s,*}$. Then:

(a) The sequence $\{x^{s,*}\}_{s\geq 1}$ is relatively compact in X.

(b) Every limit $\bar{x}$ of $\{x^{s,*}\}_{s\geq 1}$ is a pure-strategy Bayes–Nash equilibrium of the unrestricted game G.

(c) Suppose furthermore that $\pi_i \in \mathcal{C}^2$ and $f \in \mathcal{C}^1$ with Lipschitz derivatives, π_i is strictly quasiconcave in y_i, and x^* is regular.[8] Then equilibrium payoffs $U_i(x^s)$ converge at rate

$$O\big((\Delta_s^T)^2 + \Delta_s^Y\big).$$

Notice that equilibrium payoffs converge at a quadratic rate in the type space partition size (similar to Chao and Wilson, 1987; Kastl, 2012). Convergence can be demonstrated with weaker conditions such as Moroni (2022)'s "noisy observability", but they do not guarantee the existence of a discretized pure-strategy equilibrium.

It might not be possible to approximate some equilibria (e.g., those with discontinuities) and therefore, for completeness, we establish a converse to Theorem 1 to show that any sufficiently well-behaved equilibrium can be approximated.

Theorem 2 (Converse to Theorem 1)**.** *Suppose the conditions of Theorem 1(c) hold and that every $x^* \in X^*$ is continuous and monotone. Then for each $x^* \in X^*$ and for any sequence of grids with $\Delta_s^T \to 0$ and $\Delta_s^Y \to 0$, there exists a sequence of equilibria $x^{s,*} \in X^s$ of the corresponding discretized games $\{G^s\}$ such that*

$$||x^{s,*} - x^*||_\infty \to 0.$$

The result is straightforward given the assumptions, and merely highlights that conditions exist under which *any* equilibrium can be approximated via discretization.

3 Computational method

This section describes how to compute equilibria of the discretized games introduced in Section 2. We proceed in three steps. First, we restate a single-agent problem in the notation of Section 2 and discuss its discretization. Second, we introduce a recursive algorithm that computes a global optimum on the grid. Third, we extend

[8] By *regular*, we mean that the Jacobian of the first-order conditions is non-singular.

this algorithm to games and describe two equilibrium procedures: a full recursive equilibrium algorithm and a recursive iterated equilibrium algorithm.

3.1 Recursive algorithm for the single-agent problem

First, we focus on a single player i and fix the strategy of all other players, x_{-i}. We can think of this as a single-player game, $I = 1$. We refer to this as the *discrete single-agent problem*. For notational simplicity, we drop the player and grid-size indices. We denote the ordered type grid as $\mathcal{T} = \{t_1, \ldots, t_N\}$ and let $\mathcal{Y} = \{y_1, \ldots, y_K\}$ be the discrete action space for player i. Furthermore, let $\hat{\pi}_i(x, t_n) \equiv \int_{\theta \in T^s(t_n)} \pi_i(x(\theta), \theta) f(\theta)\, d\theta$ be the step-function payoff elements in equation (2).[9] Let $\hat{\pi} = \hat{\pi}_i$ in this section.

The key property that allows fast computation is the *local* structure of constraints, which we define as follows.

Definition 1. *We say that the discretized game admits an* individual local step structure *if it satisfies Assumption 1.*

Let a_n denote the candidate action at step n and $s_n = a_{n+1}$ the action at step $n + 1$. We use this notation as the action at step $n + 1$ serves as the state variable for step n in the recursion. Then, conditional on the individual local step structure, the set of admissible actions at step n conditional on s_n is given by the following correspondence:

$$C_n(s_n) \equiv \{a_n \in \mathcal{Y} : \rho_n(a_n, s_n) \leq 0\}.$$

The function $\rho(\cdot)$ is defined in Assumption 1.

We now specify our solution algorithm. We use a forward dynamic programming approach: at each step, we maximize the sum of a flow utility and a forward value (analogous to the continuation value in standard backward induction).

Algorithm 1 (Recursive single-agent algorithm)**.** *Denote by $\hat{x}$ the algorithm solution.*

- *Step 0. (Initialization) Let $V_0(\cdot) = 0$.*
- *Step 1. ($1 \leq n < N$) For each $s_n \in \mathcal{Y}$, define the Bellman equation*

$$V_n(s_n) = \max_{a_n \in C_n(s_n)} \big\{ V_{n-1}(a_n) + \hat{\pi}(a_n, t_n) \big\}.$$

[9] In applications we will typically use an appropriate approximation of the profit function $\hat{\pi}_i$, but for the sake of exposition we will leave the notation unchanged.

- *Step 2. Let $\bar{x}_n(s_n)$ be an argmax on the right-hand side of $V_n(s_n)$. If there is more than one argmax, either select one, or let the algorithm "fork out" to run the remainder of the steps separately for each fixed point. If $n < N-1$, return to step 1.*

- *Step 3. ($n = N$) Let $\hat{x}_N$ solve $\max_{a_N \in \mathcal{Y}} \{V_{N-1}(a_N) + \hat{\pi}(a_N, t_N)\}$. If there is more than one argmax, either select one, or calculate the solution for each one.*

- *Step 4. Construct the sequence $\hat{x}_1, \ldots, \hat{x}_N$ recursively via*

$$\hat{x}_n = \bar{x}_n(\hat{x}_{n+1}), \quad n = N-1, \ldots, 1.$$

Algorithm 1 requires at most $N \times K^2$ evaluations of the objective function, since at each n, K step-n actions y must be evaluated for each of K potential actions z at step-$(n+1)$. In comparison, K^N evaluations are needed for a full grid search. Despite this, the algorithm yields a *global* solution of the discrete single-agent problem whenever the local step structure holds, as the next proposition shows.

Proposition 1. *Suppose the discretized single-agent problem admits an individual local step structure. Then the set of solutions to discretized single-agent problem coincides with the set of solutions returned by Algorithm 1.*

This is a special case of Theorem 3 below, and hence the proof is omitted. Intuitively, the recursion builds, for each possible action at step $n+1$, the globally optimal "path" of actions up to step n. This implements an "intelligent grid search": dominated paths are pruned as n increases, so only paths that can be part of a global optimum survive. Mechanism design problems that satisfy the conditions discussed in Section 2.3 are thus recursively solvable in this sense.

3.2 Recursive methods for equilibrium computation

On a given grid, we seek a profile of discretized strategies $\hat{x}$ that is a (global) equilibrium of the discretized game introduced in Section 2.4. We consider two approaches. The first is a full equilibrium solution, which solves for the full global equilibrium of the game, but requires an extra assumption and is potentially computationally expensive. The second is an iterated solution that does not require any further assumptions and is generally very fast, but only yields convergence in (global) best responses.

3.2.1 Recursive full equilibrium algorithm

The first approach is a direct extension of Algorithm 1 to the multi-player environment. The single-agent problem relies on a one-dimensional assumption on local constraints for the recursive structure, and to obtain this structure in the full game, we must be able to totally order the joint type space without violating the individual local step structure for any player. We therefore state the following assumption on the type distribution.

Assumption 3. *The support of the joint type distribution, F, is such that any finite joint grid $\mathcal{T} \subset \text{supp}(F)$ can be ordered as a sequence $(t_1, \ldots, t_N)$ where, for every player i, the sequence of marginal types $(t_{i,1}, \ldots, t_{i,N})$ satisfies Assumption 1.*

This holds naturally when the type space is common for all players. A leading example is an auction in the inverse strategy formulation where players choose a quantity for each price (price plays the role of the type space in our formulation). In this case, the price space is common to all players and their strategies can be ordered accordingly.

Definition 2. *We say that a game admits a* game local step structure *if it admits an individual local step structure and furthermore, F satisfies Assumption 3. In this case, we index the types by $n = 1, \ldots, N$.*

Let $a_n = (a_{i,n}, a_{-i,n})$ denote the action profile at step n and $s_n = (s_{i,n}, s_{-i,n}) = a_{n+1}$ the profile at step $n+1$, which again will serve as a state variable. Conditional on the game local step structure then for each player i and each step $n < N$, define the admissible set

$$C_{i,n}(s_{i,n}) \equiv \left\{ a_{i,n} \in \mathcal{Y}_i : \rho_{i,n}\big(a_{i,n}, s_{i,n}\big) \leq 0 \right\}.$$

Although the underlying game is simultaneous, our algorithm constructs the equilibrium recursively step-by-step. Because players do not dynamically observe each other's choices, the recursive state space must track each player's beliefs about the joint action profile being constructed. Let $a_n^b = (a_{i,n}^b, a_{-i,n}^b)$ denote the beliefs about actions at n, and similarly let $s_n^b = (s_{i,n}^b, s_{-i,n}^b)$ denote beliefs about the state at n (which corresponds to actions at $n+1$). We now present the recursive full equilibrium algorithm.

Algorithm 2 (Recursive full equilibrium algorithm)**.** *Let $\hat{x}$ be the algorithm solution.*

- *Step 0. (Initialization) Set initial forward value:* $W_{i,0}(\cdot) = 0$ *for all players* i.

- *Step 1.* $(1 \leq n < N)$ *For each player* i *and each step-*$(n+1)$ *action* $s_{i,n}$*, define the maximized forward payoff up to step* n *as*

$$V_{i,n}(s_{i,n} \mid a_n^b) = \max_{a_{i,n} \in C_{i,n}(s_{i,n})} \left\{ W_{i,n-1}(a_{i,n} \mid a_n^b) + \hat{\pi}_i(a_{i,n}, a_{-i,n}^b, t_n) \right\}. \tag{3}$$

  Let $\bar{x}_{i,n}(s_{i,n} \mid a_n^b)$ *denote the argmax of this problem.*

- *Step 2. (Mutual belief consistency) At step* n*, conditional on the step-*$(n+1)$ *action profile* s_n*, find the mutually belief-consistent best responses* $E_n(s_n) = (E_{i,n}(s_n), E_{-i,n}(s_n))$*. These are defined implicitly as the action profiles such that every player's best response is equal to the belief about their action:*

$$E_{i,n}(s_n) \in \bar{x}_{i,n}(s_{i,n} \mid E_n(s_n)) \text{ for all } i. \tag{4}$$

  If there is more than one mutual best response, either select one, or let the algorithm "fork out" to compute each equilibrium path separately.

- *Step 3. (Value update) Carry the optimized, mutually consistent value forward to the next step. For each belief* s_n^b*, update the forward value function:*

$$W_{i,n}(s_{i,n} \mid s_n^b) = V_{i,n}\left(s_{i,n} \mid E_n(s_n^b)\right). \tag{5}$$

  If $n < N$*, set* $n \to n+1$*, and return to Step 1.*

- *Step 4.* $(n = N)$*. At step* N*, the problem is unconstrained (*$C_{i,N} = \mathcal{Y}_i$*). Solve Steps 1 and 2 without dependence on* s_n *to obtain the action profile* $\hat{x}_N$*.*

- *Step 5. (Equilibrium path) Construct the sequence of equilibrium actions* $\hat{x}_1, \ldots, \hat{x}_N$ *recursively via*

$$\hat{x}_n = E_n(\hat{x}_{n+1}), \quad n = N-1, \ldots, 1.$$

The state space in Algorithm 2 includes all possible joint actions, so the computational cost grows with the number of players. The number of evaluations of the objective function is at most $I \times N \times K^{(I+2)}$, since for each n, each of I players must evaluate K step-n actions for each of K potential actions s at step $n+1$, for each of K^{I-1} potential step-n actions of the other players, and for each of K potential beliefs about the player's action. This compares to $(K^N)^I$ evaluations for a full grid search. The former is tractable for small numbers of players and moderate grids, and has

the advantage of delivering a global equilibrium of the discretized game (and, under Theorem 1, an approximation of an equilibrium of the original game). We sometimes use interpolation to enhance the stability of the algorithm solution.

Equilibrium multiplicity can be handled by allowing the belief-consistent mutual best responses to be set-valued: each branch of the recursion then corresponds to a distinct equilibrium path. We next show that the algorithm can find all equilibria of the game.

Theorem 3. *Suppose the discrete game admits a game local step structure. Then the set of pure-strategy Bayes-Nash equilibria of the discretized game coincides with the set of solutions returned by Algorithm 2.*

The proof is in Appendix A.3.

It is important to emphasize that in applications, I represents the number of player groups, that is, groups of players who are symmetric in terms of their type distributions and strategy spaces and who are assumed to use the same strategy in equilibrium. For example, if we are looking for a symmetric equilibrium of an auction with identical bidders, then $I = 1$ regardless of the number of bidders. Hence, we can find all symmetric equilibria of the auction using at most $1 \times N \times K^{(1+2)}$ calculations for each equilibrium. Notice also that we can find equilibria that rely on certain out-of-equilibrium beliefs. For instance, auction equilibria where some bidders bid above their valuation at prices where they are sure to lose in equilibrium. Such equilibria can easily be ruled out in the algorithm, but if they are of interest, they can be solved.

3.2.2 Recursive iterated equilibrium algorithm

When the number of players or the dimensionality of the action space makes Algorithm 2 too computationally costly, we employ an iterated best-response procedure. This method preserves the tractable low-dimensional state structure of Algorithm 1 by treating opponents' strategies as fixed when solving each player's single-agent problem. Consequently, it only requires the assumption of individual local step structure, rather than the stronger game local step structure.

Algorithm 3 (Recursive iterated equilibrium algorithm)**.** *Let $x^{(j)}$ denote the strategy profile at iteration j.*

- *Step 0. (Initialization) Set $j = 1$. Initialize the algorithm with a starting strategy profile $x^{(0)}$. Choose a metric $D(\cdot, \cdot)$ on the space of strategy profiles and a tolerance level $\varepsilon > 0$.*

- *Step 1. (Best response) For each player i:*

  (a). Construct a single-agent discretized problem where the opponents' strategies are fixed at $x_{-i}^{(j-1)}$. Define the iteration-j payoff as a function of player i's candidate action $a_{i,n}$:

  $$\pi_i^{(j)}(a_{i,n}, t_n) = \pi_i\big(a_{i,n}, x_{-i,n}^{(j-1)}, t_n\big),$$

  and evaluate the constraints $\rho_{i,n}^{(j)}(\cdot,\cdot)$ given the fixed profile $x_{-i,n}^{(j-1)}$.

  (b). Solve this problem using Algorithm 1 to obtain the iteration-j candidate best-response strategy, denoted $\tilde{x}_i^{(j)}$.

- *Step 2. (Strategy update) Update the strategy profile with candidate best-response strategies. Apply a damping rule or smoothing to obtain the new profile $x^{(j)}$.*

- *Step 3. (Convergence check) Evaluate the distance between successive iterations. If $D\big(x^{(j)}, x^{(j-1)}\big) < \varepsilon$, terminate and report $x^{(j)}$ as an approximate equilibrium of the discretized game. Otherwise, set j to $j+1$ and return to Step 1.*

In our applications, we typically apply damping to reduce cycling which may occur due to the discrete grid. While Algorithm 3 is not guaranteed to converge theoretically or find all equilibria as the full equilibrium algorithm, it converges rapidly and produces accurate approximations in the environments we study. In settings where equilibrium multiplicity or unstable equilibria are central to the analysis, Algorithm 2 is preferable. With these algorithms in place, the next section tests them in a set of simulations.

4 Simulations

4.1 Asymmetric auction

We first apply our method to solve for equilibrium in asymmetric auctions. Theoretically, the equilibrium of an asymmetric auction is characterized by a boundary value problem. This problem rarely has a known closed-form solution, and therefore numerical analysis is often needed.[10]

A common approach for solving the system of differential equations in asymmetric auctions is the *shooting algorithm* (Marshall, Meurer, Richard, and Stromquist, 1994;

[10]For existence and characterization see Lebrun (2006), for a discussion of estimation with asymmetries see Lamy (2012), for a mechanism design approach see Kirkegaard (2012).

Bajari, 2001; Gayle and Richard, 2008). This method treats the problem as an initial value problem, and repeatedly solves the system of differential equations from different initial values until the solution satisfies the remaining boundary conditions. However, Fibich and Gavish (2011) show that this approach is inherently unstable, even in simple symmetric auctions, as the shooting algorithm is very sensitive to the initial condition,[11] and additional errors may accumulate along the path.

Another approach is to use *projection methods*, which approximate the solution as a linear combination of a set of basis functions (Bajari, 2001; Hubbard and Paarsch, 2009; Hubbard, Kirkegaard, and Paarsch, 2013). This approach typically uses Chebyshev polynomials as the projection basis, and solves for the parameters that minimize the errors in the differential equation subject to the boundary constraints. Yet Hubbard and Paarsch (2014) note that "a researcher cannot use these methods blindly: solutions need to be inspected to make sure they are reasonable." Below, we show that the method is indeed sensitive to initial values.

In contrast to the aforementioned approaches, our recursive method performs a global search across the entire admissible set of functions, rather than along a single path. Consequently, we avoid compounding early errors along the path and therefore will not suffer from the instabilities documented by Marshall et al. (1994) or Fibich and Gavish (2011). Furthermore, we are not subject to the traditional issues with solving systems of differential equations often used in the above methods, such as stiffness, estimation of higher-order derivatives to apply Taylor approximations and the lack of a common upper bound when there are 3 bidders or more.

The recursive method can also be applied to auction settings with interdependent values, other preferences such as risk aversion, degenerate equilibria that are not the solution of any differential equation, and to alternative auction formats such as all-pay auctions, which are difficult to solve with other numerical methods.

Model. Suppose we are in a first-price single-unit auction with I bidders and bidder i has value $v_i \sim G_i$. We use the inverse-strategy space formulation such that n indexes price and $\hat{x}_{i,n}$ is the smallest type of i who bids at price step n. The only constraint on the strategy is monotonicity: $\rho_{i,n}(\hat{x}_{i,n}, \hat{x}_{i,n+1}) = \hat{x}_{i,n} - \hat{x}_{i,n+1} \leq 0$. Hence, the game admits a game local step structure, and we can apply Algorithms 2 and 3.

[11] Consider a symmetric auction solved via backward shooting starting ϵ away from the true upper boundary condition. The solution will be $b(v) + \frac{\epsilon}{F^{n-1}(v)}$ and the second term gets big close to zero.

Examples. We use the following examples to evaluate our method's performance by comparing our solution to those of alternative numerical methods. For all cases, the existence and characterization of the equilibrium follow from Lebrun (2006).

- *Example 1.* 2 bidders, with $G_1 = G_2 = U(0, 1)$.
- *Example 2.* 2 bidders, with $G_1 = U(0, 1)$ and G_2 a mixture of a beta and uniform distribution on [0,1].
- *Example 3.* 2 bidders, with $G_1 = U(0, \frac{4}{3})$, and $G_2 = U(0, \frac{4}{5})$
- *Example 4.* 5 bidders, with $G_1 = G_2 = G_3 = U(0, 1)$ and $G_4 = G_5 = U(0, \frac{3}{4})$
- *Example 5.* 3 bidders, with $G_1 = G_2 = U(0, 10)$ and $G_3 = U(2, 8)$

We present two sets of results: Table 1 compares results from our iterated equilibrium algorithm to a set of existing numerical approaches that a researcher may apply in each of the examples; Table 2 presents results of our full equilibrium algorithm on the same examples. Thus, in Table 1 the recursive solution refers to Algorithm 3, whereas Table 2 the recursive solution refers to Algorithm 2.

Iterated equilibrium algorithm results. In Table 1, K is the number of grid points in the grid for the action (the smallest valuation bidding at step), and N is the number of grid points in the grid for the type (the price).[12] We smooth win probabilities at each iteration to eliminate discontinuities due to the discreteness of opposing bidders' bid strategies.[13] The iterated approach uses a stopping criterion that must balance accuracy of the result with computation time. While loose tolerance for the stopping criteria yield poor predictions, too tight thresholds can cause cycling behavior due to the discrete approximation, increasing runtime without meaningfully improving results. We set the tolerance so that win probabilities at each price move by less than 1 percent at the final step. At each iteration, the current player picks the bid that maximizes expected profit given the current win probability function, a vector giving the probability of winning at each price on the grid. New win probabilities are

[12]The reported compute times were calculated on a machine with Intel(R) Xeon(R) Gold 6242R CPU @ 3.10GHz, max clock speed 3093, 130 GB RAM, running Windows 11, using MATLAB R2023a.

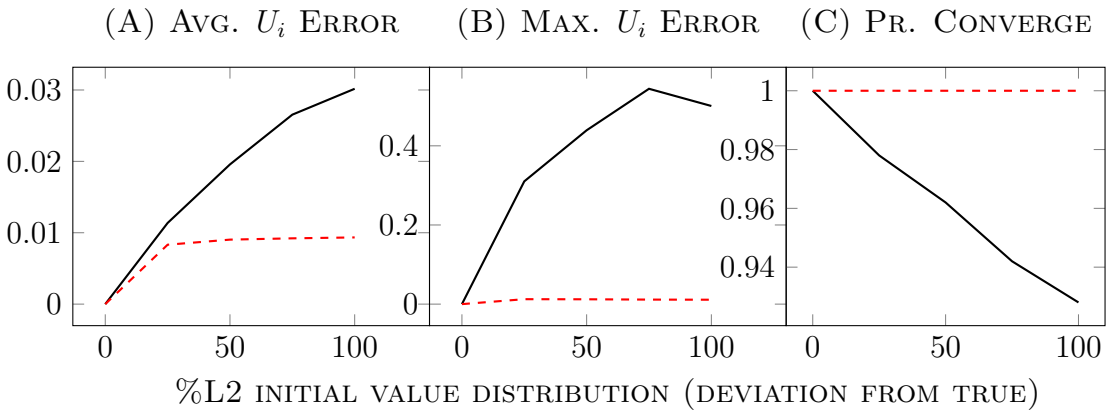
[13]The results are broadly similar without smoothing, however the convergence tolerance may need to be set higher and the impact of increases or decreases in the number of grid points on the number of iterations-to-converge can be hard to predict. This is likely due to differences between the constrained strategic equilibrium and the Bayes Nash equilibrium of the unconstrained game.

Table 1: Asymmetric Auction Iterated Solution – Simulation

	Profit		Compute
	Bidder 1	Bidder 2	Time (s)
Example 1			
Analytic	0.1667	0.1667	-
Shooting	0.1666	0.1666	43.69
Polynomial ($M = 10$)	0.1664	0.1664	3.13
Polynomial ($M = 20$)	0.1419	0.1477	1.31
Polynomial ($M = 30$)	0.1657	0.1657	1.86
Recursive ($K = 5, N = 100$)	0.1683	0.1683	0.16
Recursive ($K = 10, N = 100$)	0.1668	0.1668	0.34
Recursive ($K = 20, N = 200$)	0.1727	0.1727	0.52
Recursive ($K = 30, N = 200$)	0.1688	0.1688	0.92
Example 2			
Shooting	0.1016	0.2276	36.06
Polynomial ($M = 10$)	0.1013	0.2273	1.20
Polynomial ($M = 20$)	0.0995	0.2257	1.31
Polynomial ($M = 30$)	0.1013	0.2273	1.64
Recursive ($K = 5, N = 100$)	0.1067	0.2546	0.74
Recursive ($K = 10, N = 100$)	0.1010	0.2185	1.25
Recursive ($K = 20, N = 200$)	0.1030	0.2214	6.14
Recursive ($K = 30, N = 200$)	0.1039	0.2197	8.84
Example 3			
Analytic	0.3110	0.0980	-
Recursive ($K = 5, N = 100$)	0.3134	0.0926	0.33
Recursive ($K = 10, N = 100$)	0.3061	0.0890	0.48
Recursive ($K = 20, N = 200$)	0.3109	0.0919	2.19
Recursive ($K = 30, N = 200$)	0.3051	0.0906	3.81
Example 4			
Polynomial ($M = 10$)	0.0382	0.0162	0.27
Polynomial ($M = 20$)	0.0423	0.0215	1.01
Polynomial ($M = 30$)	0.0280	0.0043	2.39
Recursive ($K = 5, N = 100$)	0.0589	0.0110	0.10
Recursive ($K = 10, N = 100$)	0.0525	0.0098	0.30
Recursive ($K = 20, N = 200$)	0.0529	0.0104	1.95
Recursive ($K = 30, N = 200$)	0.0526	0.0103	2.35
Example 5			
Recursive ($K = 5, N = 100$)	0.9387	0.5381	0.29
Recursive ($K = 10, N = 100$)	0.8433	0.4611	0.73
Recursive ($K = 20, N = 200$)	0.8708	0.4899	2.40
Recursive ($K = 30, N = 200$)	0.9096	0.5029	2.50

Note: *Shooting* and *Polynomial*: Examples 1 and 2 use Hubbard and Paarsch (2014) via the *SNOPT* package; M denotes the number of Chebyshev terms. Polynomial in Example 4 uses *fmincon* and Hubbard and Kirkegaard (2019) analytics for the support tail. *Recursive*: Recursive Iterated Equilibrium Algorithm. *Analytic*: analytical solution. Initial values: truthful bidding. Tolerance: $\epsilon = 0.005$.

Figure 2: Auction Initial Value Sensitivity



computed from the resulting strategies, then dampened toward the old values (10% new, 90% old) to prevent cycling, and repeated until convergence. We next discuss the examples in turn.

Example 1 can be solved analytically, and in this simple setting, all three methods obtain accurate solutions.[14] Our solution requires a similar computation time to the polynomial approach, and is much faster than the shooting algorithm. Figure 2 illustrates the sensitivity of the polynomial approximation to initial values with a Chebyshev polynomial with $M = 20$. We take random perturbations of the initial values, $\beta_0 = d \times (\frac{r}{\sum r})$, where β_0 are the parameters describing the true solution, d is a constant chosen by the researcher and r is a vector of uniform random draws. Computing the equilibrium for each one, we find that, when comparing the polynomial solution to our recursive solution, average profit errors expand more quickly, maximum errors expand much faster, and the convergence proportion drops.

Example 2 introduces asymmetry into the auction.[15] Since the value distributions have common supports, the polynomial and shooting approaches can again be used to provide a benchmark. The patterns are similar to those discussed above in Example 1. While the shooting algorithm performs well in both Example 1 and 2, Hubbard and Paarsch (2014) and Fibich and Gavish (2011) present cases where the shooting method fails to obtain the correct solution.

The next examples move to cases where the support of bidders' value-distributions differ across bidder-groups. Example 3 illustrates a special case which admits an analytic solution. For Example 4, no analytic solution is available, but Hubbard and Kirkegaard (2019) propose a method for numerically solving the group-symmetric

[14] This corresponds to Example 1 of Hubbard and Paarsch (2014).
[15] This corresponds to Example 2 of Hubbard and Paarsch (2014).

Table 2: Asymmetric Auction Full Solution – Simulation

	Profit		Compute
	Bidder 1	Bidder 2	Time (s)
Example 1			
Recursive ($K = 5, N = 50$)	0.1695	0.1695	2.70
Recursive ($K = 5, N = 100$)	0.1742	0.1742	3.13
Recursive ($K = 5, N = 200$)	0.1792	0.1792	5.96
Recursive ($K = 10, N = 50$)	0.1722	0.1722	8.47
Recursive ($K = 10, N = 100$)	0.1738	0.1738	17.18
Recursive ($K = 10, N = 200$)	0.1759	0.1759	34.31
Example 2			
Recursive ($K = 5, N = 50$)	0.1272	0.2562	1.64
Recursive ($K = 5, N = 100$)	0.1327	0.2550	3.42
Recursive ($K = 5, N = 200$)	0.1387	0.2617	6.68
Recursive ($K = 10, N = 50$)	0.1174	0.2394	8.98
Recursive ($K = 10, N = 100$)	0.1168	0.2369	18.20
Recursive ($K = 10, N = 200$)	0.1195	0.2397	36.48
Example 3			
Recursive ($K = 5, N = 50$)	0.3225	0.0794	1.49
Recursive ($K = 5, N = 100$)	0.3269	0.0799	3.05
Recursive ($K = 5, N = 200$)	0.3354	0.0800	6.06
Recursive ($K = 10, N = 50$)	0.3168	0.0910	8.86
Recursive ($K = 10, N = 100$)	0.3222	0.0905	17.20
Recursive ($K = 10, N = 200$)	0.3263	0.0912	34.74
Example 4			
Recursive ($K = 5, N = 50$)	0.0561	0.0108	1.43
Recursive ($K = 5, N = 100$)	0.0568	0.0101	2.93
Recursive ($K = 5, N = 200$)	0.0560	0.0099	5.72
Recursive ($K = 10, N = 50$)	0.0550	0.0110	8.38
Recursive ($K = 10, N = 100$)	0.0555	0.0110	17.07
Recursive ($K = 10, N = 200$)	0.0559	0.0107	33.08
Example 5			
Recursive ($K = 5, N = 50$)	1.1517	0.7365	17.04
Recursive ($K = 5, N = 100$)	1.1369	0.6965	34.18
Recursive ($K = 10, N = 50$)	0.8907	0.5489	207.77
Recursive ($K = 10, N = 100$)	0.9133	0.5570	411.90

Note: The examples correspond to the previous table. *Recursive*: the Recursive Full Equilibrium Algorithm.

equilibrium. They derive a mechanical link between the maximum bids of the two groups that allows the researcher to worry about only the smaller maximum bid and to solve a standard problem with a single boundary condition. Their polynomial method suffers from the same concerns with stability, as the example in Figure 2. Example 5 is from Dharanan and Ellis (2024).[16] It is difficult to gauge the accuracy of the solution obtained using their approach, for which they can provide only bounds that allow them to conclude the maximum error in the bid functions are bounded above by roughly 0.5. Their approach requires substantial computation to calculate the solution even for this example. Our method obtains a solution, which resembles their approximated solution in seconds.

We observe the following about the performance of the step-function algorithm relative to other methods: the algorithm (a) produces stable output for different specifications; (b) scales well, so that computation time increases slowly with number of asymmetric groups; (c) is fast; (d) works over a wide range of auction examples without any need to adapt the code; (e) extends easily to complex settings.[17] In particular, the computational time required is similar to that of the polynomial approach of Hubbard and Paarsch (2014). However, by not relying on a differential equation representation, the method extends far more easily to more complex settings, such as Dharanan and Ellis (2024), where it achieves far better performance than existing numerical techniques. Unlike tools based on solving the differential equations, which require the researcher to provide specialized knowledge and application specific tailoring, the step function approach handles these difficult examples with no user-input.

Full equilibrium algorithm results. In Table 2, we show the results for Algorithm 2. We use payoff interpolation to enhance the stability of the results.

First, on computation times. Computation cost increases slowly in N, but faster in K. The computation times scale in a fairly predictable and constant manner across Examples 1-4, which is expected as they involve similar numbers of calculations. In Example 5, computation times are much slower, as we have 3 agent types (for illustration, we do not allow the code to take advantage of the symmetry of Types 1 and 2). This illustrates the substantial additional costs as the number of players expands in the full solution approach.

Second, on accuracy. In Examples 1 and 3, where analytic solutions exist, the

[16] It corresponds to their Example 3.

[17] Indeed, all 5 examples in this section were computed with exactly the same code, which takes as input only the number of bidders, the distribution of valuations, and grid sizes.

maximum profit error over the grid configurations in the table is 0.01-0.02, which corresponds to 6-7% in Example 1 and for bidder 1 in Example 3, whereas it corresponds to 20% for bidder 2 in Example 3, where the bidder profit is smaller. Notice that in the latter case, increasing K to 10 brings the relative error to under 10%.

4.2 Nonlinear pricing

In this section, we apply our method to nonlinear pricing games. Our approach accommodates competition, multidimensional types, and type-specific participation constraints - features that are difficult to handle with standard methods. Nonlinear tariffs are widely used for efficiency improvements and price discrimination in settings such as regulated utilities, airlines, banking, and consumer goods, and are also central to analyses of optimal taxation and public goods.

We focus on competition in schedules, where computational challenges have limited empirical work. Existing approaches either impose strong structure —- such as increasing hazard rates to ensure first-order conditions are sufficient (Luo, Perrigne, and Vuong, 2018) – or rely on quasi-linear preferences that yield tractable linear demand (Miravete, 2002; Aryal and Gabrielli, 2020).

By contrast, our method requires minimal assumptions on preferences, incorporates all local constraints (both downward and upward), and computes the global optimum directly by exploiting the recursive structure of the constraints and profit function, rather than relying on local or parametric approximations.

Model. Consider the following model of nonlinear pricing. There are two firms, $i = 1, 2$. Consumers have a quadratic utility function

$$u(x_1, x_2, t) = \frac{\alpha + t}{\beta - \gamma}(q_1 + q_2) - \frac{\beta}{2(\beta^2 - \gamma^2)}(q_1^2 + q_2^2) - \frac{\gamma}{\beta^2 - \gamma^2} q_1 q_2 - p_1 q_1 - p_2 q_2,$$

where $x_i = (p_i, q_i)$ are the price and quantity of the bundle of goods they acquire from firm i, whereas $t \sim U(0, 10)$ is their type. We again formulate the problem in discrete space with local constraints. Firm i offers contracts $\hat{x}_{i,n} = (\hat{q}_{i,n}, \hat{p}_{i,n}) \in \mathcal{Y}^K$ to types $t_n \in \mathcal{T}^N$, with the two-dimensional contract grid being $\mathcal{Y}^K = ((p_1, q_1), \ldots, (p_K, q_K))$. The firm has profit function $\pi_i(\hat{x}_{i,n}, t_n) = \hat{q}_{i,n}(\hat{p}_{i,n} - c)$. The set of local constraints, upward and downward, at step n, is

$$\rho_{i,n}(\hat{x}_{i,n}, \hat{x}_{i,n+1}) = \begin{cases} u(\hat{x}_{i,n}, \hat{x}_{-i,n}, t_n) - u(\hat{x}_{i,n+1}, \hat{x}_{-i,n}, t_n) \geq 0; \\ u(\hat{x}_{i,n+1}, \hat{x}_{-i,n+1}, t_{n+1}) - u(\hat{x}_{i,n}, \hat{x}_{-i,n+1}, t_{n+1}) \geq 0. \end{cases}$$

We then solve for equilibrium using Algorithm 3, and let D be given by the difference between subsequent iterations of the individual rationality constraint $u(\cdot)$.

Simulations. This section presents results from two examples. We set $\gamma = -\frac{1}{3}$, so the products are complements. In addition, $\alpha = 1$, $\beta = 1 + \gamma$. Firms have constant marginal costs of production $c = 2$. The set-up matches the delegated common agency model of Martimort and Stole (2009), and this setting has a simple analytic solution.

- *Example 1.* Fix one firm to play the analytic solution. Use the utility offered from that firm's menu for every consumer type as a type-specific participation constraint. Then solve the optimization problem for the remaining firm.
- *Example 2.* Solve the competitive model where both firms play optimally.

Table 3 compares our approximation, a polynomial approximation, and the true analytic solution for these examples.[18] In Example 1 (monopoly), the recursive method is more accurate, faster, and does not suffer from sensitivity to starting values, unlike the polynomial method. The same pattern holds in Example 2 (competition), where we use a recursive iterated equilibrium algorithm that does require starting values, but the results are insensitive to them, in contrast to the polynomial solver. The recursive method achieves high accuracy with $K = 10$ and $N = 100$ and is still faster than any polynomial solver configuration.

The gap in computational time between the polynomial approach and our method is larger here than in the auction. While the auction's error calculations rely on simple matrix operations, the nonlinear pricing problem requires determining consumer choices for each price schedule, which introduces additional nonlinearities and increases the cost of evaluating errors for the polynomial approximation. Figure 3 illustrates the recursive equilibrium solution for Example 2.

Multidimensional nonlinear pricing. Multidimensional settings are substantially more complex. In the parametric utility approach, the inability to integrate by parts leads to a second-order calculus-of-variations *obstacle problem*; the standard strategy is to solve a relaxed problem and verify the remaining constraints ex post (Rochet and Choné, 1998; Deneckere and Severinov, 2017). Alternatively, Ghili and Yoon (2023) use a linear approximation to maximize a non-smooth objective with no

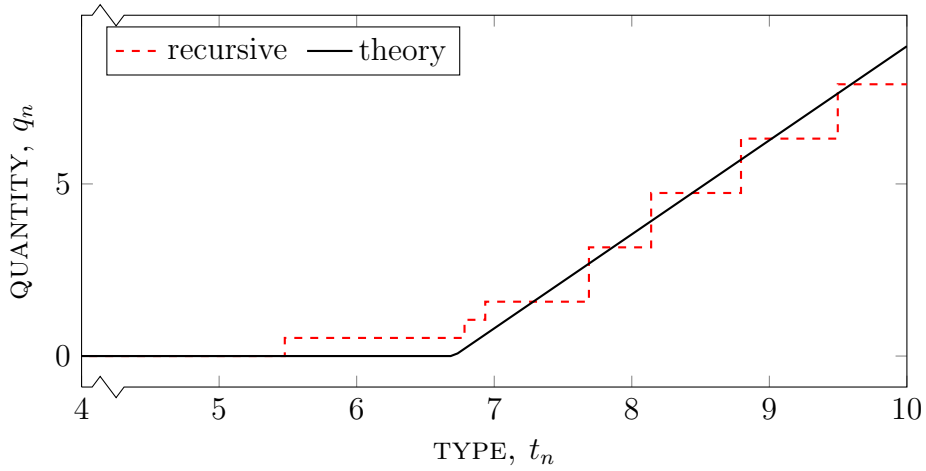
[18] The polynomial approach is discussed in Section 4.1.

Table 3: Nonlinear Pricing – Simulation

	Profit		Compute
	Bidder 1	Bidder 2	Time (s)
Example 1			
Analytic	5.65	-	-
Polynomial ($M = 1$, SV = true)	5.61	-	2.19
Polynomial ($M = 12$, SV = true)	5.28	-	4.21
Polynomial ($M = 1$, SV = far)	5.08	-	1.49
Polynomial ($M = 12$, SV = far)	3.84	-	5.09
Recursive ($K = 8$, $N = 100$)	5.54	-	0.06
Recursive (K =10, $N = 100$)	5.47	-	0.07
Recursive (K =20, $N = 200$)	5.64	-	1.04
Example 2			
Analytic	5.65	5.65	-
Polynomial (M= 1, SV = true)	5.88	5.62	5.58
Polynomial (M=12, SV = true)	5.57	4.12	88.90
Polynomial (M= 1, SV = far)	5.87	5.00	14.10
Polynomial (M=12, SV = far)	5.00	4.39	48.80
Recursive ($K = 8$, $N = 100$)	5.04	5.04	2.93
Recursive ($K = 8$, $N = 100$, SV = true)	5.04	5.04	1.90
Recursive ($K = 8$, $N = 100$, SV = far)	5.04	5.04	2.30
Recursive ($K = 10$, $N = 100$)	5.51	5.51	2.56
Recursive ($K = 10$, $N = 100$, SV = true)	5.51	5.51	3.43
Recursive ($K = 10$, $N = 100$, SV = far)	5.51	5.51	5.05
Recursive ($K = 20$, $N = 200$)	5.58	5.58	47.50
Recursive ($K = 20$, $N = 200$, SV = true)	5.58	5.58	46.87
Recursive ($K = 20$, $N = 200$, SV = far)	5.58	5.58	93.62

Note: In the competitive case we explore the role of starting values: SV changes the starting values. Here, *true* indicates the starting values are the true solution, while *far* starts the solution from [1,-0.01] for the linear example and a vector of ones for the richer B-spline approximation. *Polynomial*: our own implementation of the polynomial method. *Recursive:* the Recursive Single-Agent Algorithm for Example 1 and the Recursive Iterated Equilibrium Algorithm for Example 2. Notice that K refers to the grid size of both price and quantity, such that the combined grid has a size of K^2. Notice that in Example 1, the recursive algorithm has no starting values as it is solving a single-agent problem.

FIGURE 3: COMPETITIVE NONLINEAR PRICING – ONE-DIMENSIONAL TYPES



guarantees on concavity or linearity, relying on a grid-bisection method whose convergence to the global maximum is not ensured. Rochet and Stole (2003) propose checking binding local constraints via dynamic programming, but the path-dependency of utility prevents a tractable representation of the firm's optimal price schedule.

We study a multidimensional agent-type problem in which the firm has a one-dimensional screening instrument—the price for q units. Unlike methods that parametrize and approximate $p(q)$ directly, our approach extends naturally to richer contracting environments.

To evaluate performance in multidimensional settings where the true solution is often intractable, we consider an artificial experiment with a known solution. We focus on a single seller from Martimort and Stole (2009), fixing opponents' behavior at their best responses. Agents draw two-dimensional types (ω_1, ω_2), with the sum $t = \omega_1 + \omega_2$ as the payoff-relevant type, uniformly distributed. The algorithm is not told that there is only one true type; it must solve the full two-type problem.

Solving the full optimal schedule here is infeasible: for instance, an iterative approach moving along the horizontal axis would require a state space depending on all vertical grid points at each step. Instead, we solve a relaxed problem and iteratively add constraints. We define a grid of type-regions and move recursively in a snake-like pattern, storing the last state and the current constraint set. Starting with no constraints, we solve the unconstrained problem in one full recursive pass. Then, in each column, we add as a constraint the cell with the largest incentive-constraint violation (if any). We repeat this procedure.

In many cases the solution has few violations, yielding a reasonable approximation at low cost. In practice, we stop after the second iteration because the number of

constraints becomes too large. This approximate solution yields profits of 6.41 versus the true 5.71. Although we illustrate the method for a single agent, the same iterative structure can be applied to the competitive one-dimensional example.

5 Application: procurement auctions

A large literature has used asymmetric auctions models to evaluate procurement policies. Many applications use the alternative numerical approaches discussed in Section 4, which are numerically unstable and can be computationally slow. This limits the set of counterfactuals that can be analyzed, which could be important for cases such as reserve price determination, subsidies for entry, changes to the supply of contracts such as pooling auctions together, to name a few. We examine two applications and illustrate the increases in accuracy, decreases in model restrictions required and computational gains from using our recursive approach.

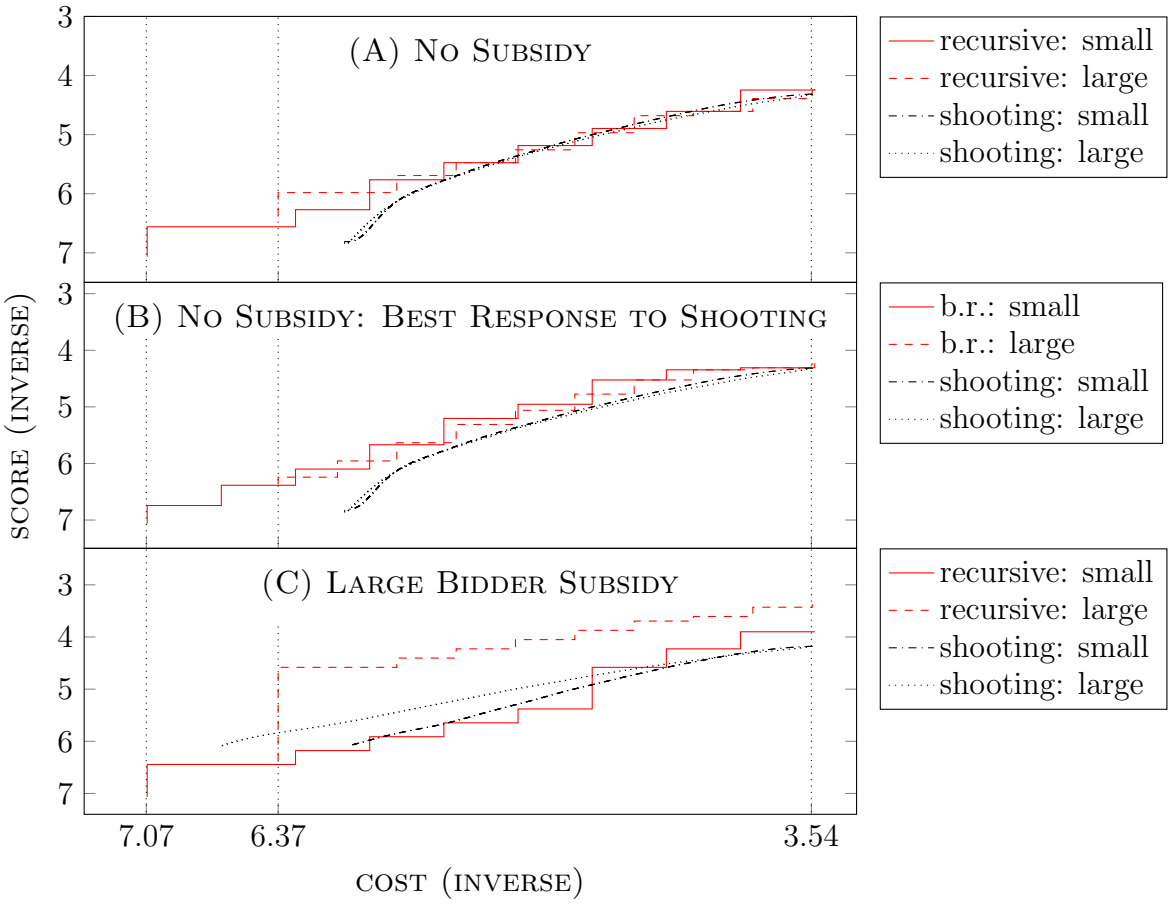
5.1 California bid preference program

Bid preference programs are widely used across the U.S., allowing some bidders to pay only a fraction of their bid and thereby introducing asymmetry and uncommon supports even when bidders are initially similar. These features complicate boundary conditions, especially for shooting algorithms used to evaluate counterfactuals: while two-bidder boundaries are typically straightforward, multi-bidder settings can exhibit "bid-separation" (Hubbard and Kirkegaard, 2019), which the literature has largely ignored. Many studies instead impose left boundary conditions that rule out separation (e.g. Marion, 2007; Hubbard and Paarsch, 2009; Krasnokutskaya and Seim, 2011), potentially understating competition, particularly with several strong bidders.

To examine the consequences of this restriction, we revisit the California Small Business Preference program in Caltrans highway and street maintenance auctions, as studied by Krasnokutskaya and Seim (2011), focusing on their first auction (project ID 1177). Our method highlights cases where bid separation could arise and substantially alter profit predictions, while cutting the average equilibrium-computation time by over 90%. Moreover, we show settings where the shooting algorithm fails to converge and stops short of the right boundary, whereas our method always reaches the boundary; both bid separation and incomplete convergence can significantly affect bidder profits.

Figure 4 compares our recursive method with the shooting algorithm of Kras-

Figure 4: Bid Strategies – Preference Program

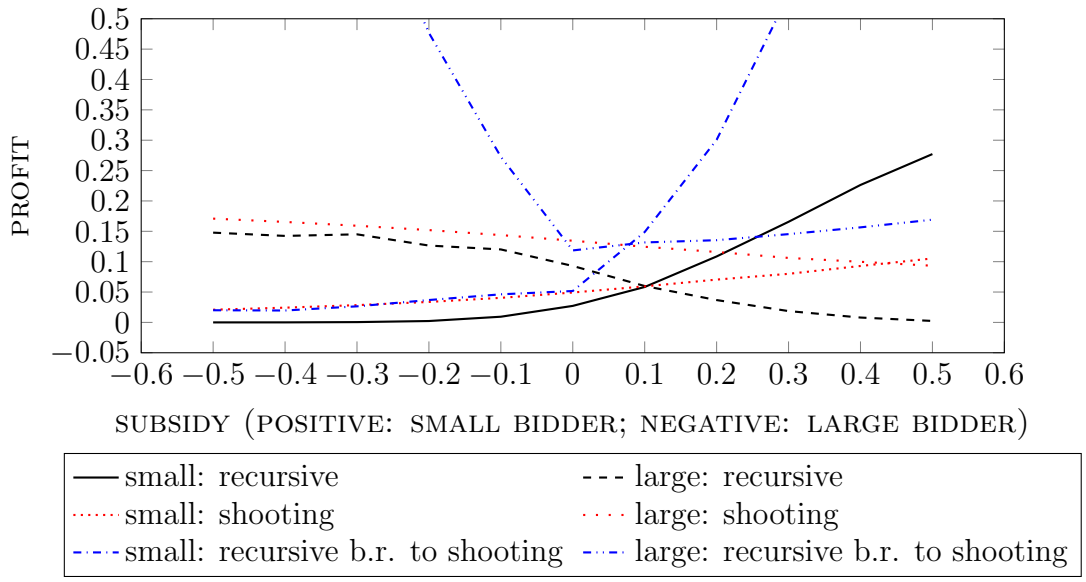


nokutskaya and Seim (2011) in an auction with 3 large bidders (costs on $[3.54, 6.37]$) and 2 small bidders (costs on $[3.52, 7.07]$). Multiple large bidders and differing cost supports make bid separation theoretically possible. Panel A plots equilibrium bids under no subsidies, with inverted axes to match the standard auction representation. The shooting algorithm "undershoots", failing to reach the left boundaries, in line with failures documented in Fibich and Gavish (2011); by construction, the recursive algorithm does not suffer from this issue. Panel B shows the recursive algorithm's best response to the shooting equilibrium from Panel A. The recursive best response differs and yields higher profits (as shown below).

Panel C shows the same auction with a 30% subsidy to large bidders[19] This increases asymmetry and the recursive solution induces features bid separation: large bidders continue to bid where small bidders have dropped out. This is visible on the right side of the figure, where the recursive strategy for large bidders ends above that

[19]Bids are shown in stated dollars; actual payments for large bidders are 1.3 times higher.

Figure 5: Bidder Profits – Subsidy Variation



of small bidders (corresponding to more aggressive, lower bids). Separation is absent in the shooting solution, which imposes a boundary condition that rules it out.

To illustrate the economic implications of the different algorithm outcomes, Figure 5 compares bidder profits under varying subsidies to large bidders in the same auction. Dotted and loosely dotted lines show the shooting-algorithm solution of Krasnokutskaya and Seim (2011), while solid and dashed lines show ours. The results reveal substantial differences: small (respectively, large) bidder profits can be markedly underestimated (resp., overestimated) when small bidders receive a subsidy.

We also plot dash-dotted and dash-dot-dotted lines depicting profits from the recursive best response against opponents playing the shooting equilibrium. These indicate that, for most subsidy levels, a bidder can profitably deviate from the shooting strategy. In a few cases the recursive best response yields slightly lower profits, likely because it is constrained to a coarser strategy space and subject to numerical integration error; elsewhere, it delivers large profit gains.

5.2 Michigan highway procurement auctions

We next apply our method to the Michigan highway procurement auctions analyzed by Somaini (2020). They feature interdependent values, allowing us to evaluate our method outside the private values framework. The Michigan Department of Transport uses first-price sealed-bid auctions to award construction and maintenance contracts that pay a fixed amount to the winning firm, regardless of realized cost. Somaini

Table 4: Uniform Private Values – Algorithm Comparison

	Expected Profit				Compute Time (s)		
			Recursive			Recursive	
N	Analytic	Shooting	(1)	(2)	Shooting	(1)	(2)
2	0.1667	0.1531	0.1746	0.1619	4.49	0.07	9.45
3	0.0833	0.0734	0.0850	0.0824	15.90	0.10	20.55
4	0.0500	0.0418	0.0534	0.0487	29.13	0.18	41.78
5	0.0333	0.0263	0.0342	0.0269	61.05	0.33	17.22
6	0.0238	0.0218	0.0231	0.0217	96.48	0.97	40.61

Note: *Shooting*: The shooting algorithm of Somaini (2020). *Recursive*: the Recursive Iterated Equilibrium Algorithm. Version (1) is the standard implementation of the algorithm, version (2) is implemented with Somaini (2020)'s cost function code. The grid parameters are $K = 10$ and $N = 100$.

(2020) approximates the solution with a shooting algorithm and models cost as a linear function of observable project characteristics and bidders' (privately observed) signals. Thus, costs are interdependent and bidding is subject to the winner's curse.

To test our approach, we introduce a perturbed cost function that moves us away from the baseline linear-normal case:

$$c_i = \delta c_p + (1 - \delta) l_i (c_w s_i)^{c_e},$$

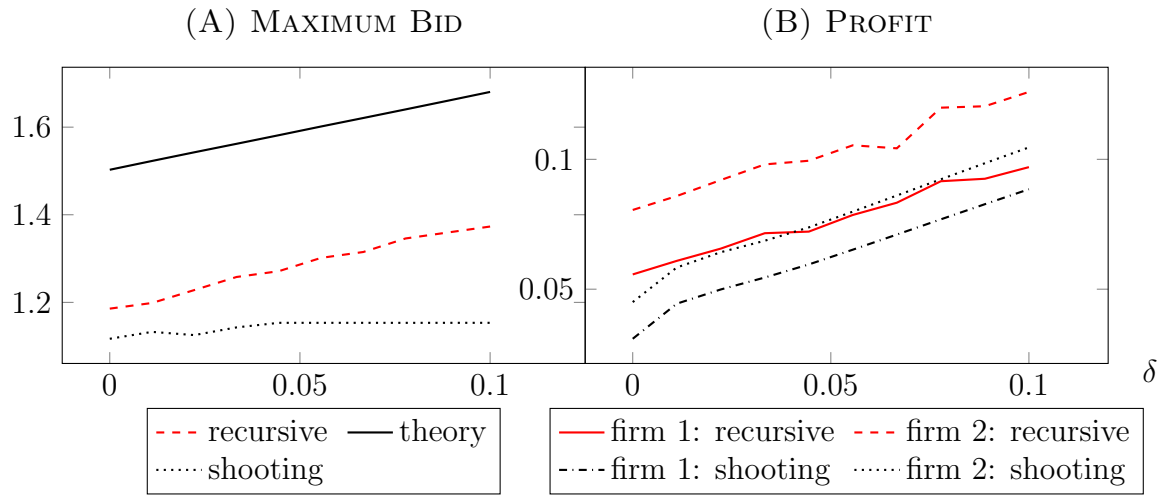
where c_p the costs from Somaini (2020)'s model, δ, a weight between his cost and the additional component, s_i is the signal in quantile space, l_i and c_w are linear coefficients on the bidders' own signal, and c_e induces curvature. This specification nests examples 1, 3 and 4 from our simulation exercise, (e.g. for example 1, set $\delta = 0$ and $l_i = c_w = c_e = 1$), and the original model (set $\delta = 1$). We anticipate that the nonlinearity will complicate convergence compared to the original model.

We first measure baseline performance in a symmetric auction with values on $U(0, 1)$. Table 4 compares the speed and accuracy of our method to that of Somaini (2020)'s shooting algorithm, for different numbers of bidders, N. Our method appears to provide accurate approximation of bidder profits and scales better. It maintains computational efficiency, while the shooting algorithm slows down significantly as the number of bidders increases. For context, Somaini (2020) reports that all auctions for the Michigan procurement auction sample with fewer than 10 bidders were solved in less than 5 hours. This length limits the scope for the counterfactual analysis.[20]

Finally, Figure 6 explores the sensitivity of both approaches to perturbations of

[20]Somaini (2020) overcomes this by using a subsample of 250 auctions and restricted participation probabilities, thus avoiding some of the most computationally intensive areas of the parameter space.

FIGURE 6: EVALUATING CONVERGENCE – PERTURBED DGP



the form described above. We select a random auction from the data, and fix $c_e = 2$ and $c_w = 2$, so that the quadratic term is more 'aggressive' than the linear. We use the values of l_i from the locations of firms in the data. To remove effects associated with the number of bidders, we set the number of bidders at $N = 2$. Panel A shows that the maximum bid of the shooting algorithm undershoots its theoretical level more than our method, and the difference increases as a function of δ. Panel B highlights that these differences are empirically relevant, with the approximated profits from the shooting algorithm on average only 75% of those from our method.

6 Application: cereal product sizes

We apply our method to study the menu of box sizes offered by cereal firms, using demand estimates obtained using NielsenIQ retail scanner and consumer panel data. The application relates to recent policy concern over "shrinkflation", and negative press coverage of cereal companies selling smaller boxes.[21] While these concerns could reflect increases in market power, menu adjustments could also increase consumer welfare; banning the practice may leave consumers worse off. Despite periodic public concern – similar concerns were raised in 2007[22] – cereal companies have regularly

[21] Congress introduced a bill prevent this practice: `https://www.casey.senate.gov/imo/media/doc/shrinkflation_report.pdf`. See also McNair (2023) or `https://www.weforum.org/agenda/2022/03/how-companies-are-hiding-inflation-without-charging-you-more/`.

[22] See `https://www.forbes.com/2007/08/27/general-mills-update-markets-equity-cx_af_0827markets28.html?sh=65aa1c5524c3`.

adjusted prices and package sizes to market conditions over the last decades.

Existing empirical work documents how firms across many industries adjust their product line characteristics and variety due to competition (Berry and Waldfogel, 2001; McManus, 2007; Borzekowski, Thomadsen, and Taragin, 2009; Watson, 2009; Sweeting, 2010; Seim and Viard, 2011) and price discrimination (Draganska and Jain, 2005, 2006). However, solving the counterfactual optimal set of varieties is challenging, because the number of possible menus is exponential in the number of potential products.

To manage this challenge, the existing literature typically restricts firms to a small, discrete set of choices and solves the simplified game (Fan and Yang, 2020; Wollmann, 2018; Eizenberg, 2014; Draganska et al., 2009). While this approximation might be reasonable in some contexts, it is not likely to perform well in cereal markets, where firms offer numerous menu options and package weights might vary by a fraction of an ounce. Using standard optimization tools in these settings would require nesting a complex optimization problem at each step of the best-response iteration (e.g. Fan, 2013).

Our method simplifies this optimization considerably, allowing for the number of points to be endogenous while capturing both horizontal and vertical differentiation. To maintain tractability, we make two assumptions on the demand model. First, that package specific shocks occur at the product level rather than the product-size level; adding an additional size does not give the firm an additional draw of the consumer taste. Second, menu adjustments and consumer preferences occur along a single dimension (e.g. package size).

In the case of cereals, the simultaneous design of 25 product lines is too high dimensional to effectively compute using the full solution approach, and we will solve for equilibrium iteratively as in Section 4.2. However, in a setting with fewer product lines to be simultaneously calculated, the full solution approach could also be applied.

6.1 Data and background

The cereal market is characterized by frequent changes in size offerings. To summarize the changes between 2018 and 2022, we divide product sizes into three groups: small (below 12 oz), medium (12-18 oz) and large (18-24 oz).[23] In 2022, small boxes are offered by 50 % of the products, down from 62% in 2018, and the average size of small boxes grows from 6.91 oz to 7.71 oz. In the same period, all products were offered in

[23]We drop the single-serve market, with boxes less than 5 ounces, as these are not close substitutes.

a medium sized format, and the average size of a median box shrank from 14.97 oz to 14.72 oz. In the large package category the trend appears to be the opposite. Finally, in 2022 only 42% of products are sold in large boxes, down from 71% in 2018, and the average size of those that are offered, increases from 20.38 oz to 20.89 oz per box.

6.2 Demand estimation

We specify the following random coefficients logit model where consumer i's utility for product j of size k in market t is

$$u_{ijkt} = \beta p_{jkt} + \beta_i size_{jk} + \omega_j + \xi_{jkt} + \epsilon_{ijkt}, \tag{6}$$

where p_{jkt} is product price, $size_{jk}$ is product size, ω_j is a brand-level fixed effect, ξ_{jkt} captures market product-size preferences, and ϵ_{ijkt} is an error term. The model has a random coefficient on size only, which we specify as

$$\beta_i \sim \mathcal{N}(\Pi d_i, \Sigma\Sigma'), \tag{7}$$

where d_i is demographics and we estimate the parameters Π and Σ.

Estimation follows Backus, Conlon, and Sinkinson (2021) and Nevo (2000b).[24] We use data from 5 designated market areas (DMAs): Boston, Chicago, Charlotte, Denver, and Richmond. We aggregate unit sales and revenues to the DMA-channel-week level and focus exclusively on the set of conventional supermarket sales (which Nielsen labels as "F" stores).

To make counterfactual predictions, we must estimate preferences and costs for sizes not observed in the data. We linearly interpolate preferences ξ_{jkt} between observed points and make them flat outside the range. Marginal costs are modeled as a quadratic function $mc(q) = a_1 + a_2 q + a_3 q^2$. We fit this function to the unrestricted estimates of marginal costs and restrict the marginal cost at the grid's upper bound so that the largest package the firm wants to offer is the one they offer in the data. Finally, we estimate bounds on the fixed cost for each package size, by using our method to compute the firm's optimal menu against the opponents' current menus, and then finding the package fixed cost that equates the number of packages in the optimal menu and the data.

[24] We set the outside market share following the average share from Backus et al. (2021). We include micro moments for the covariances of package size with income and with a kids indicator, as well as the average income and average kids indicator, all conditional on purchase.

6.3 Evaluating model fit

To evaluate the model fit, we benchmark it by solving equilibrium in a single randomly selected market in September of 2018. We assume a single ϵ_{ijt} rather than ϵ_{ijkt} to ensure that firms do not introduce additional package sizes only to gain an additional draw of the idiosyncratic shock.

The model approximates the observed menus closely, in terms of availability of sizes (measured in percentage of firms offering the size) and average size, as summarized in Table 5 below. It also approximates profits well, although it slightly overshoots the small-package profits. The computed menus also use too many steps with 2.96 steps on average, as opposed to 2.12 in the data.

Table 5: Predicted vs. Observed Menus (Sep 2018)

	Availability		Average Size (oz)	
Category	Predicted	Data	Predicted	Data
Small (< 12 oz)	64%	60%	9.94	10.55
Medium (12–18 oz)	88%	76%	15.96	15.20
Large (18–24 oz)	76%	52%	21.42	20.93

The model also does a good job predicting cross-product variation: Firm A typically offers a menu with three options, whereas Firm B has higher variance in both products count and size. Firm D offers only 2 product sizes but with some variation in their small and large box sizes.

Because the iterated approach to jointly optimizing the menus could find local optima, we tested whether different product orderings arrive at the same solution (e.g. orderings based on consumer product taste or firm costs). All orderings arrived to the same solution as the random initial ordering. We also tested local stability by perturbing opponents' inclusive-value utility by per-type random shocks with L_1 norm of 0.05, 0.25, and 1, redrawn 100 times at each magnitude, and recomputing the flagship product of Firm A's optimal menu. At every perturbation magnitude tested, the optimal menu was unchanged i.e., the recomputed (price, quantity) bundles fell on exactly the same grid points as the baseline.

6.4 Counterfactuals

We now use the model to evaluate how firms adjust menus in response to two structural shocks: a production cost increase, and a horizontal merger. We compare our

results to a setting with fixed box sizes to emphasize the importance of considering quantity adjustments.

Cost Increases. We evaluate a 10% increase in production costs, for all firms, and track the resulting changes in sizes and prices of the boxes offered. As costs increase, firms tend to trim their product-line, and the average number of offerings falls from 2.84 to 2.56. Now, 56% of firms offer a small box, down from 64% before with the average small box size increasing to 10.11oz. The size of medium boxes goes down slightly to 15.5 oz on average from 15.96 oz and 84% of firms offer a medium box from 88% in the baseline. In addition, fewer firms offer a large box (56%, down from 76%) and the average size conditional on non-empty falls slightly to 20.88 oz from 21.42 oz. This suggests a consolidation toward the medium menu, rather than uniform changes across all categories.

Allowing for the box size adjustment also affects welfare, with quantity changes in general softening the blow for firms. As described in Table 6, all firms except Firm D have higher profits under flexible sizes than under fixed sizes, with very heterogeneous effects across firms. This suggests that ignoring the adjustment of box-sizes understates the level of competition in the market and the ability of firms to respond to a cost shock. Restricting the ability of firms to adjust box sizes could also be bad for consumers, as consumer surplus experiences a much steeper drop when sizes are fixed than when they are flexible.

Table 6: Welfare Impacts of Cost Increase: Flexible vs. Fixed Sizes

	Flexible Sizes (% Change)	Fixed Sizes (% Change)
Consumer Surplus	−5.4	−7.5
Firm Profits:		
Firm A	−6.0	−9.6
Firm B	−5.0	−5.1
Firm C	+1.0	−2.2
Firm D	−13.0	−9.0

These results suggest that box size adjustments are an important margin for firms facing cost increases. In addition, restricting the ability of firms to adjust the size of their boxes as proposed in the current discussion about combating shrinkflation could be bad for consumers.

Hypothetical Merger. We now analyze two hypothetical mergers (following Nevo, 2000a) and investigate the effect of box size adjustments in this context. Table 7 below shows the changes in profits from two hypothetical mergers.

Table 7: Profit Changes from Merger

	Flexible Sizes (% Change)	Fixed Sizes (% Change)
Merger: D and A		
Consumer Surplus	−2.2	−4.3
Firm Profits:		
A	+30.1	+27.3
B	+2.1	+3.1
C	+12.1	+4.1
Merger: D and B		
Consumer Surplus	−3.0	−5.0
Firm Profits:		
A	+4.8	+4.7
B	+17.2	+18.5
C	+14.1	+5.7

Note: *Fixed Quantity baseline* computed with pyBLP single-unit choice model at BLP-equilibrium prices (fitted costs). *Adj. Quantity baseline* computed from the nonlinear pricing equilibrium, where consumers choose both product and quantity from the menu. The two frameworks compute CS (consumer surplus) from different models of consumer choice, so levels differ; only percentage changes are comparable across columns.

Following a merger of firms A and D, A's profits rise by more than the direct effect of D's pre-merger profits, with modest gains for B's and important gains for C. When box sizes are held fixed, the gains for firm C are smaller. Post-merger, the availability of large boxes decreases (from 76% to 68%) and the average size of those remaining falls from 21.43 oz to 20.89 oz. The small and medium box sizes offered by firms are relatively similar.

Following a merger of firms B and D, it is again the merged firm (B in this case) that has the largest profit increase, and C has significantly smaller profits when box sizes are held fixed. The trends in box sizes are similar to the merger with A. However there is some growth in the average size of a medium box, from 15.19 oz to 15.39 oz. The merged firm expands its medium sized offerings and cuts back on the large box sizes offered for D's former products. Finally, the estimated loss in consumer surplus is approximately twice as large when box sizes are fixed, as when they are flexible,

suggesting that box size adjustments are generally beneficial to consumers.

7 Conclusion

This paper introduces a novel method for computing counterfactual equilibria in static games with private information. By approximating complex functions – such as bid functions in auctions or tariffs in price discrimination – with finite-dimensional step functions, we can take advantage of the local constraints structure of many of these games to formulate a sequential program. This program can be solved by dynamic programming techniques, allowing us to find the *global solution* in the set of step functions rapidly, avoiding local equilibria and the numerical instabilities that traditional shooting algorithms and polynomial approximations are subject to.

We demonstrate the method's robustness in different settings. First, we show in a simulation exercise that our method avoids the typical instabilities of shooting algorithms and polynomial approximations in asymmetric auctions and nonlinear pricing settings, and furthermore we show how our method can be applied to multidimensional settings. Second, we reexamine two applications to procurement auctions and show that unlike existing approaches our method can capture issues such as bid separation, and we provide significant speed improvements that make new counterfactual analysis possible.

Finally, we examine the ready-to-eat cereal market where we can analyze counterfactual menu adjustments on a full grid of quantities and prices, unlike the previous literature. We show that taking into account quantity adjustments is important for policy recommendations. Following the recent discussion of shrinkflation, we consider a counterfactual 10% increase in production costs and show that the full model has different predictions for profits and consumer surplus than a model where firms are only allowed to adjust prices.

Appendix A Proofs

A.1 Proof of Theorem 1

Proof of Theorem 1. We now prove the different parts of the theorem.

Part (a): relative compactness. By Assumption 2, π_i satisfies weak single-crossing and weak quasisupermodularity. By Reny (2011), these conditions ensure that any discretized game G^s has a pure-strategy Bayes-Nash equilibrium $x^{s,*}$ where for each i, $x_i^{s,*}$ is monotone. Furthermore, since the action space Y_i is compact, the sequence of functions $\{x_i^{s,*}\}$ is uniformly bounded.

Since $x_i^{s,*}$ is measurable, monotone and uniformly bounded for all s, then by Helly's Selection Theorem extended to vector-valued functions, there exists a subsequence of $\{x_i^{s,*}\}$ that converges pointwise everywhere to a measurable and monotone limit function $\bar{x}$. As the action space Y_i is compact, the limit also maps into this space: $\bar{x}_i : T_i \to Y_i$. Finally, the constraint function R_i is continuous, and therefore the constraints are preserved in the limit. Hence, $\bar{x} \in X$, and therefore the subsequence is relatively compact.

Part (b): limit is Bayes-Nash equilibrium. Suppose for contradiction that $\bar{x}$ is a limit point of $\{x^{s,*}\}$ and $\bar{x} \notin X^*$.

The payoff function π_i is bounded, and hence the equilibrium payoffs converge to some limit vector $\bar{u} = \lim_{s\to\infty} U(x^{s,*})$. Because the unrestricted game G is better-reply secure, there exists a player i, a deviation $x_i' \in X_i$, an $\epsilon > 0$, and a neighborhood V of $\bar{x}_{-i}$, such that for all $y_{-i} \in V$:

$$U(x_i', y_{-i}) > \bar{u}_i + \epsilon.$$

Since X_i^s is dense in X_i in the limit, and because G being better-reply secure implies G being payoff secure, there is a sequence of actions $x_i'^s \in X_i^s$ converging to x_i', such that for sufficiently large s,

$$U_i(x_i'^s, y_{-i}) > \bar{u}_i + \frac{\epsilon}{2} \text{ for all } y_{-i} \in V.$$

Since $x_{-i}^{s,*} \to \bar{x}_{-i}$ pointwise almost everywhere, there exists an integer S such that for $s > S$, $x_{-i}^{s,*} \in V$. Thus, for $s > S$,

$$U_i(x_i'^s, x_{-i}^{s,*}) > \bar{u}_i + \frac{\epsilon}{2}.$$

Since $x^{s,*}$ is an equilibrium of the discretized game G^s, then $U_i(x^{s,*}) \geq U_i(x_i'^s, x_{-i}^{s,*})$. Taking the limit as $s \to \infty$ gives $\bar{u} \geq \bar{u} + \epsilon/2$, a contradiction. Thus $\bar{x} \in X^*$.

Part (c): convergence rate. Let $B : X \to X$ be the best-response mapping of the unrestricted game, such that $x^* = B(x^*)$. Conversely, $B^s : X^s \to X^s$ is the best-response mapping of the discretized game, so that $x^{s,*} = B^s(x^{s,*})$.

Let $\tilde{B}(x)$ be the best response in step functions on $(T_{i,n}^s)_{n=1}^{N_i^s}$ where the action is unrestricted, i.e. is in Y_i. Call this the *intermediate mapping*. Since $B(x)$ is continuous and monotone, $\|\tilde{B}(x) - B(x)\|_\infty = O(\Delta_s^T)$. The difference between the discrete mapping B^s and the intermediate mapping $\tilde{B}$ is bounded by the projection error: $\|B^s(x) - \tilde{B}(x)\|_\infty \leq (1/2)\Delta_s^Y = O\big(\Delta_s^Y\big)$. By the previous results and the triangle inequality,

$$\|B^s(x) - B(x)\|_\infty = O\big(\Delta_s^T + \Delta_s^Y\big). \tag{A.1}$$

Let $G(x) \equiv x - B(x)$. Since $\pi_i \in \mathcal{C}^2$, B is continuously differentiable, and we can take a first-order Taylor expansion of $G(x^{s,*})$ around x^*:

$$G(x^{s,*}) = G(x^*) + \nabla G(x^*) \cdot (x^{s,*} - x^*) + o(\|x^{s,*} - x^*\|_\infty). \tag{A.2}$$

By definition, $G(x^*) = 0$. Since x^* is regular, the Jacobian of the first-order condition is non-singular and therefore $\nabla G(x^*)$ is invertible. Let $C = [\nabla G(x^*)]^{-1}$. Finally, $G(x^{s,*}) = x^{s,*} - B(x^{s,*}) = B^s(x^{s,*}) - B(x^{s,*})$. For sufficiently large s, the higher-order terms are dominated, and hence (A.1) and (A.2) yield:

$$\|x^{s,*} - x^*\|_\infty \leq \|C\|_\infty \cdot \|B^s(x^{s,*}) - B(x^{s,*})\|_\infty = O\big(\Delta_s^T + \Delta_s^Y\big). \tag{A.3}$$

Take a second-order Taylor expansion of U_i around the continuous equilibrium x^*:

$$\begin{aligned} U_i(x^{s,*}) - U_i(x^*) = &\int_T \nabla_{x_i} \pi_i(x^*(\theta), \theta) \cdot (x_i^{s,*}(\theta) - x_i^*(\theta)) f(\theta) d\theta \\ &+ \int_T \nabla_{x_{-i}} \pi_i(x^*(\theta), \theta) \cdot (x_{-i}^{s,*}(\theta) - x_{-i}^*(\theta)) f(\theta) d\theta \\ &+ O(\|x^{s,*} - x^*\|_\infty^2) \\ =&0 + O(\Delta_s^Y) + O\big((\Delta_s^T + \Delta_s^Y)^2\big) = O\big((\Delta_s^T)^2 + \Delta_s^Y\big). \end{aligned}$$

In the second equality, the first term follows by the Envelope Theorem, the second term by the Midpoint Rule which cancels the linear error term in the type space, and the third term by substituting (A.3).

□

A.2 Proof of Theorem 2

Proof of Theorem 2. Fix a regular, continuous and monotone equilibrium $x^* \in X$.

As in Theorem 1, let $B : X \to X$ be the best-response mapping of the unrestricted game, such that $x^* = B(x^*)$. Also, let $B^s : X^s \to X^s$ denote the best-response mapping of the discretized game, so that $x^{s,*} = B^s(x^{s,*})$. Finally, let V denote an arbitrarily small neighborhood around x^*.

From the proof of Theorem 1(c), the best response distance is uniformly bounded:

$$||B^s(x) - B(x)||_\infty = O\big((\Delta_s^T)^2 + \Delta_s^Y\big).$$

Therefore, as $\Delta_s^T, \Delta_s^Y \to 0$, we have $||B^s - B||_\infty \to 0$. Because x^* is regular (i.e. the Jacobian evaluated at x^* is non-singular), we can apply the Implicit Function Theorem for perturbed operators. This assures that for sufficiently high s (where $||B^s - B||_\infty$ is sufficiently small), B^s has a fixed point $x^{s,*} \in X^s$ in V, which by definition is an equilibrium of the discretized game. As this is true for any arbitrarily small V, there must exist a sequence $\{x^{s,*}\}$ with $||x^{s,*} - x^*||_\infty \to 0$. □

A.3 Proof of Theorem 3

Standard backward induction does not apply to this problem because the actions are chosen simultaneously at each step, and constraints link adjacent steps. Therefore, the algorithm utilizes a forward dynamic programming approach. To resolve the simultaneity, we construct an *augmented state variable* that includes the players' beliefs about concurrent actions. We are now ready to prove Theorem 3.

Proof of Theorem 3. First, notice that for a sequence of beliefs $x^b = (x_1^b, \ldots, x_N^b)$ that satisfy the constraints, player i's optimization problem over strategy x_i can be described as a finite-horizon deterministic dynamic program with augmented states $(s_{i,n}, s_n^b)$, where $s_n^b = x_{n+1}^b$ is the belief about the action at step $n+1$. By the Principle of Optimality, the recursive maximization in Step 1 and the value update in Step 3 of the algorithm identify the global maximum of the expected forward payoff $U_i(x_i, x_{-i}^b)$. Consequently, any sequence constructed from the argmax correspondence $\bar{x}_{i,n}$ is a global best response to the belief path x^b.

Part 1. A solution to the algorithm must be an equilibrium strategy.

Suppose the algorithm produces a solution path $\hat{x}$. By Step 2 of the algorithm, the solution enforces mutual belief-consistency at each step n, such that the belief path satisfies $a_n^b = \hat{x}_n$. Then, by Step 5, the sequence $(\hat{x}_1, \ldots, \hat{x}_N)$ is constructed recursively from the mutually consistent best responses, meaning $\hat{x}_{i,n} = E_{i,n}(\hat{x}_{n+1}) \in \bar{x}_{i,n}(\hat{x}_{i,n+1} \mid \hat{x}_n)$. By the preliminary argument, $\hat{x}_i$ is a global best response to $\hat{x}_{-i}$. Since this holds for all players i, $\hat{x} = x^*$ where x^* is a pure-strategy Bayes-Nash equilibrium of the discretized game.

Part 2. An equilibrium strategy must be a solution to the algorithm.

Let x^* be a pure-strategy Bayes-Nash equilibrium of the discretized game.

Evaluate the recursive algorithm along the equilibrium belief path where $a_n^b = x_n^*$ and $s_n^b = x_{n+1}^*$ for all n. By definition, x_i^* is a global best response to x_{-i}^* for all i. Therefore, by the Principle of Optimality, the sequence of actions x_i^* must satisfy the recursive Bellman equations at each step n. Hence, at each step n, $x_{i,n}^* \in \bar{x}_{i,n}(x_{i,n+1}^* \mid x_n^*)$, and therefore $x_n^* = E_n(x_{n+1}^*)$. Consequently, we can select x^* as a valid path when constructing the sequence recursively in Step 5, yielding $x^* = \hat{x}$.

Parts 1 and 2 together show that the set of solutions to the algorithm is equal to the set of pure-strategy Bayes-Nash equilibria of the discretized game.

□

References

Aryal, G. and M. F. Gabrielli (2020): "An Empirical Analysis of Competitive Nonlinear Pricing," *International Journal of Industrial Organization*, 68, 102538.

Backus, M., C. Conlon, and M. Sinkinson (2021): "Common Ownership and Competition in the Ready-to-Eat Cereal Industry," NBER Working Papers 28350, National Bureau of Economic Research.

Bajari, P. (2001): "Comparing Competition and Collusion: A Numerical Approach," *Economic Theory*, 18, 187–205.

Berry, S. T. and J. Waldfogel (2001): "Do Mergers Increase Product Variety? Evidence from Radio Broadcasting," *Quarterly Journal of Economics*, 116, 1009–1025.

Borzekowski, R., R. Thomadsen, and C. Taragin (2009): "Competition and price discrimination in the market for mailing lists," *Quantitative Marketing and Economics*, 7, 147–179.

Carroll, G. (2012): "When Are Local Incentive Constraints Sufficient?" *Econometrica*, 80, 661–686.

Chao, H.-P. and R. Wilson (1987): "Priority Service: Pricing, Investment, and Market Organization," *American Economic Review*, 77, 899–916.

Deneckere, R. and S. Severinov (2017): "A Solution to a Class of Multi-Dimensional Screening Problems: Isoquants and Clustering," *mimeo*.

Dharanan, G. and A. Ellis (2024): "Asymmetric auctions: Perturbations, ϵ-equilibrium, and equilibrium," *Games and Economic Behavior*, 147, 1–18.

Draganska, M. and D. C. Jain (2005): "Product-Line Length as a Competitive Tool," *Journal of Economics & Management Strategy*, 14, 1–28.

——— (2006): "Consumer Preferences and Product-Line Pricing Strategies: An Empirical Analysis," *Marketing Science*, 25, 164–174.

Draganska, M., M. Mazzeo, and K. Seim (2009): "Beyond Plain Vanilla: Modeling Joint Product Assortment and Pricing Decisions," *Quantitative Marketing and Economics*, 7, 105–146.

Eizenberg, A. (2014): "Upstream Innovation and Product Variety in the U.S. Home PC Market," *Review of Economic Studies*, 81, 1003–1045.

Fan, Y. (2013): "Ownership Consolidation and Product Characteristics: A Study of the US Daily Newspaper Market," *American Economic Review*, 103, 1598–1628.

Fan, Y. and C. Yang (2020): "Competition, Product Proliferation, and Welfare: A Study of the US Smartphone Market," *American Economic Journal: Microeconomics*, 12, 99–134.

Fibich, G. and N. Gavish (2011): "Numerical Simulations of Asymmetric First-price Auctions," *Games and Economic Behavior*, 73, 479–495.

Gayle, W.-R. and J. F. Richard (2008): "Numerical solutions of asymmetric, first-price, independent private values auctions," *Computational Economics*, 32, 245–278.

Ghili, S. and R. Yoon (2023): "An Empirical Analysis of Optimal Nonlinear Pricing," *mimeo*.

Gonzalez-Eiras, M., J. Kastl, and J. Rüdiger (2023): "How Much is Being a Primary Dealer Worth? Evidence from Argentinian Treasury Auctions," *mimeo*.

Hubbard, T. and H. Paarsch (2014): "On the Numerical Solution of Auction Models with Asymmetries within the Private Values Paradigm," in *Handbook of Computational Economics*, ed. by K. Schmedders and K. L. Judd, Elsevier, 38–112, section: 2.

Hubbard, T. P. and R. Kirkegaard (2019): "Bid-Separation in Asymmetric Auctions," *mimeo*.

Hubbard, T. P., R. Kirkegaard, and H. J. Paarsch (2013): "Using economic theory to guide numerical analysis: Solving for equilibria in models of asymmetric first-price auctions," *Computational Economics*, 42, 241–266.

Hubbard, T. P. and H. J. Paarsch (2009): "Investigating Bid Preferences at Low-price, Sealed-bid Auctions with Endogenous Participation," *International Journal of Industrial Organization*, 27, 1–14.

Kastl, J. (2012): "On the properties of equilibria in private value divisible good auctions with constrained bidding," *Journal of Mathematical Economics*, 48, 339–352.

Kirkegaard, R. (2012): "A Mechanism Design Approach to Ranking Asymmetric Auctions," *Econometrica*, 80, 2349–2364.

Krasnokutskaya, E. and K. Seim (2011): "Bid Preference Programs and Participation in Highway Procurement Auctions," *American Economic Review*, 101, 2653–86.

Lamy, L. (2012): "The Econometrics of Auctions with Asymmetric Anonymous Bidders," *Journal of Econometrics*, 167, 113–132.

Lebrun, B. (2006): "Uniqueness of the Equilibrium in First-price Auctions," *Games and Economic Behavior*, 55, 131–151.

Luo, Y., I. Perrigne, and Q. Vuong (2018): "Structural Analysis of Nonlinear Pricing," *Journal of Political Economy*, 126, 2523–2568.

Marion, J. (2007): "Are Bid Preferences Benign? The Effect of Small Business Subsidies in Highway Procurement Auctions," *Journal of Public Economics*, 91, 1591–1624.

Marshall, R. C., M. J. Meurer, J.-F. Richard, and W. Stromquist (1994): "Numerical Analysis of Asymmetric First Price Auctions," *Games and Economic behavior*, 7, 193–220.

Martimort, D. and L. Stole (2009): "Market Participation in Delegated and Intrinsic Common-Agency Games," *RAND Journal of Economics*, 40, 78–102.

McManus, B. (2007): "Nonlinear Pricing in an Oligopoly Market: the Case of Specialty Coffee," *RAND Journal of Economics*, 38, 512–532.

McNair, K. (2023): "Getting Less for the Same Price? How the CPI Measures "Shrinkflation" and its Impact on Inflation," Beyond the Numbers: Prices and Spending Vol. 12 no. 2, Bureau of Labor Statistics.

Miravete, E. J. (2002): "Estimating Demand for Local Telephone Service with Asymmetric Information and Optional Calling Plans," *Review of Economic Studies*, 69, 943–971.

Moroni, S. (2022): "Existence of trembling hand perfect and sequential equilibrium in stochastic games," *mimeo*.

Nevo, A. (2000a): "Mergers with Differentiated Products: The Case of the Ready-to-Eat Cereal Industry," *RAND Journal of Economics*, 31, 395–421.

——— (2000b): "A Practitioner's Guide to Estimation of Random-Coefficients Logit Models of Demand," *Journal of Economics & Management Strategy*, 9, 513–548.

Reny, P. J. (2011): "On the existence of monotone pure-strategy equilibria in Bayesian games," *Econometrica*, 79, 499–553.

Reny, P. J. and S. Zamir (2004): "On the Existence of Pure Strategy Monotone Equilibria in Asymmetric First-Price Auctions," *Econometrica*, 72, 1105–1125.

Rochet, J.-C. and P. Choné (1998): "Ironing, Sweeping, and Multidimensional Screening," *Econometrica*, 66, 783–826.

Rochet, J.-C. and L. A. Stole (2003): "The Economics of Multidimensional Screening," *Econometric Society Monographs*, 35, 150–197.

Seim, K. and V. B. Viard (2011): "The Effect of Market Structure on Cellular Technology Adoption and Pricing," *American Economic Journal: Microeconomics*, 3, 221–51.

Somaini, P. (2020): "Identification in Auction Models with Interdependent Costs," *Journal of Political Economy*, 128, 3820–3871.

Sweeting, A. (2010): "The Effects of Mergers on Product Positioning: Evidence from the Music Radio Industry," *RAND Journal of Economics*, 41, 372–397.

Watson, R. (2009): "Product Variety and Competition in the Retail Market for Eyeglasses," *The Journal of Industrial Economics*, 57, 217–251.

Wollmann, T. G. (2018): "Trucks without Bailouts: Equilibrium Product Characteristics for Commercial Vehicles," *American Economic Review*, 108, 1364–1406.