# Efficient Grid Search to Solve Static Games with Private Information*

Jakub Kastl[†] Eric Richert[‡] Jesper Rüdiger[§]

February 2025

**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 the objective function, e.g. the bid function in an auction, can often be approximated arbitrarily well by a piecewise constant function. 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 Monte Carlo study of asymmetric auctions and nonlinear pricing games suggests that the method is stable, fast, and easily extends to complex games that are difficult to solve with existing methods. We then examine two applications to highway procurement auctions and show that our method leads to increases in accuracy and speed, and has fewer required model restrictions. Finally, we use our method to shed light on recent concerns of shrinkflation. We study the breakfast cereal market and compute globally incentive-compatible counterfactual package size and price adjustments following a cost shock.

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

*We thank 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, Dep. of Economics

[‡]University of Chicago, Dep. of Economics

[§]Universidad Carlos III de Madrid, Dep. of Business Administration

# 1 Introduction

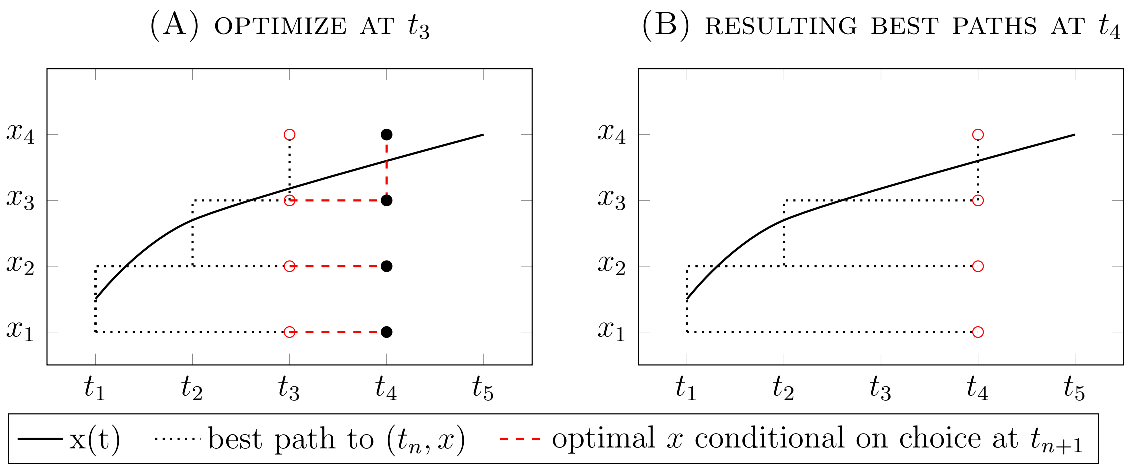
In many games, such as auctions, nonlinear pricing, optimal contracts and menu design, the equilibrium is a functional fixed point, and this is typically hard to solve. Yet we often want to solve the game, for example, to make counterfactual predictions on the effect of subsidies, mergers, technology changes or adjustments to market design. A common approach is to solve for an approximation to the solution using numerical methods. A simple way to do this would be to approximate the solution on a discrete grids, and then to do a 'brute force' full grid search of all combinations to find the global solution. However, with even a small number of grid points, this becomes computationally intractable. Therefore, standard solution methods either rely on more restrictive parameteric forms or on path-dependence from a differential equation representation of the problem. While this may lead to computational savings relative to a naive grid search, the solutions are often unstable and sensitive to initial values. Furthermore, they often lack flexibility 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 problems the objective function, such as a tariff function in price discrimination games or a bid function in auctions, can be arbitrarily closely approximated by a piece-wise constant (i.e., a step) function. Second, in a large class of games with 'local constraints', which includes a vast majority of existing solution methods, the agents' optimization problem 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 the following way. First, we conduct a Monte Carlo study to benchmark against existing methods. We show that our recursive algorithm (a) accurately approximates profits; (b) is fast; (c) scales well, so that computation time increases slowly with the number of agents; (d) works over a wide range of auction examples without any need to adapt the code; (e) extends easily to complex settings. We next reexamine two studies of highway procurement auctions that rely on existing methods, and show that our recursive algorithm increases accuracy, decreases model restrictions required 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 on prices and package sizes. Existing methodologies typically limit the feasible menu choices to make the analysis tractable, but this is not reasonable in the cereal market (and many other markets) as large variety exists in products and menu choices. Our method allows to solve the simultaneous menu design of 25 product lines by 4 firms with an endogenous number of price-quantity pairs per product. We next discuss in greater detail the method and our results.

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. To illustrate, suppose that we are in a single-unit auction in which a bidder submits a bid which is monotonically increasing in her valuation, and we wish to solve for this bidder's optimal (inverse) strategy. We can think of the problem as follows. At each step $t_n$ on the price grid we choose the lowest valuation $x(t_n)$ which bids at this price, with

FIGURE 1: OPTIMIZATION WITH RECURSIVE ALGORITHM




$x(t_n) \leq x(t_{n+1})$. At the first price step, $t_1$, for each potential $x(t_2)$ we can calculate our optimal choice of $x(t_1)$. At the second price step, for each potential $x(t_3)$ we can calculate our optimal choice of $x(t_2)$, knowing what our choice of $x(t_1)$ is conditional on $x(t_2)$. And so forth. Therefore, at step $n$ we (i) take into account the effect of our choice at $n$ on optimal choices at steps $n' < n$, and (ii) we condition on our (hypothetical) choice at step $n+1$. This allows us to find a global solution very efficiently.[1]

We illustrate the recursive algorithm in Figure 1 for a single bidder's optimal inverse bid strategy. Our objective is to approximate the inverse strategy function $x(t)$, represented by the solid line, on the grids of valuations, $x_1, ..., x_4$, and prices, $t_1, ..., t_5$. In panel A of the figure, we are at price step $t_3$. The filled circles represent the potential $x(t_4)$ and the empty circles the potential $x(t_3)$. The dotted lines represent the best paths leading up to each of the potential $x(t_3)$, i.e. the best choices of $x(t_1)$ and $x(t_2)$, conditional on $x(t_3)$. The dashed lines represent the optimal choice of $x(t_3)$, conditional on each $x(t_4)$. Thus, for instance if $x(t_4) = x_2$, then $x(t_3) = x_2$ is optimal, whereas if $x(t_4) = x_4$, then $x(t_3) = x_3$ is optimal. In panel B, we see the resulting best paths leading up to each potential $x(t_4)$. Notice that in panel A there is a path that leads to $x(t_3) = x_4$, but no path in panel B goes through this point, as it is eliminated by the optimal choices at the previous step. Hence, 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 approach finds the global solution as would a full grid search, but allows us to move much faster through the grid.

To compute the pricing game equilibrium, where opponents' strategies can also vary, we extend the single-agent recursive 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. Although this approach will find the global solution, the state space of the dynamic programming problem grows at a rate equal to the number of players. Second, the *recursive iterated equilibrium algorithm* avoids in-

[1]We show in Section 2 that in the single-agent problem our recursive algorithm requires $N \times K^2 + N$ calculations whereas a full grid search requires $K^N$ calculations.

creasing 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 Monte Carlo experiments.

We next describe the results of our Monte Carlo study in more detail. 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. This allows us to provide a comparison of the performance of our solution method to well-known numerical methods for solving differential equations, such as shooting algorithms and collocation methods. 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. In the simple examples, we show that in terms of speed, our recursive iterated equilibrium algorithm outperforms 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, and it becomes unstable for certain values for the polynomial size parameter, whereas our method's solution is stable for a wide range of parameter values. In the more complex examples, our method has comparable accuracy and similar or better solution times than the specialized problem-specific algorithms of Hubbard and Kirkegaard (2015) and Dharanan and Ellis (2024). The recursive full equilibrium algorithm provides similar results, but is naturally slower, particularly as grid sizes are increased.

Second, we apply the method to solve for the equilibrium in the nonlinear pricing game of Martimort and Stole (2009). In the nonlinear pricing context, we can solve the problem either by using the demand profile approach, or by solving for a set of contracts specifying a price and quantity for each consumer type, that incentivizes truthful reporting. 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. To model competition in this environment, we propose an approach iterating the individual rationality constraint. 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 collocation, 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 again show how iteration can be used to reduce the dimension of the state-space.

We then turn to applications. First, we examine two applications to highway procurement auctions. In particular, we apply our method to Krasnokutskaya and Seim (2011) and to Somaini (2020). We illustrate the increases in accuracy, decreases in model restrictions required and computational gains from using our method. In particular, Krasnokutskaya and Seim (2011) analyze the California Small Business Preference program which gives preference to small firms in procurement auctions, thus creating (further) asymmetry between bidders. They use a shooting algorithm to solve for the equilibrium. In such settings, when bidders are asymmetric and there are more than two bidders, bid bifurcation is possible (Hubbard and Kirkegaard, 2015). However, bid bifurcation is ruled out by the boundary condition imposed in most empirical studies, including Krasnokutskaya and Seim (2011). In addition 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 we analyze an auction where our method computes an equilibrium that allows for bid bifurcation, and is not subject to the instabilities from the shooting method.

Somaini (2020) analyzes Michigan highway procurement auctions that feature interdependent costs and also solves counterfactuals using a shooting algorithm. The interdependent costs introduce a winner's curse into the bidding process, and potentially make it more difficult to solve for equilibrium. We first benchmark his version of the shooting algorithm by calculating the equilibrium of an auction with private uniform values on [0,1] for different numbers of bidders. Consistent with the results of our Monte-Carlo, our method is more accurate and faster than the shooting based solution. The larger the number of bidders, the higher the speed advantage of our recursive method. Given that Somaini (2020) reports some auctions with high runtimes of the shooting algorithm (especially with many bidders), and equilibrium must be computed many times for many counterfactuals of interest, this speed advantage is important. We then explore the performance of the shooting algorithm of Somaini (2020) under alternative cost functions. In particular, we perturb the linear cost function from Somaini (2020) by introducing a quadratic term to test how well the methods handle this increased complexity. We show that in this case, the paper's shooting algorithm does not reach the upper bound of the signal distribution (as predicted by Fibich and Gavish (2011)), whereas by definition our recursive algorithm always reaches this bound. Furthermore, the maximum bids of the shooting algorithm are also significantly below the theoretical maximum bid, whereas our recursive algorithm is slightly below, and this distance becomes smaller as we select a finer grid for approximating bid strategies. We thus argue that our recursive method seems to converge better, and we show that this can generate significant differences in profit estimates between our recursive algorithm and the shooting algorithms used in the literature.

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. The standard assumptions on preferences often used in studies of menu design (e.g. Fan and Yang (2020)) imply that local constraints guarantee global incentive compatibility. This is not exploited in the existing literature, but since our recursive method is able to incorporate all local constraints, this implies that we can obtain a globally incentive compatible solution to the single-firm menu optimization problem. Then, we can use our iterated solution algorithm, which nests this problem, to compute counterfactual menu adjustments. This was difficult to incorporate using existing tools due to the combinatorial nature of the full grid search. To explore the importance of this, 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 the adjustment of box-sizes understates the level of competition in the market and the ability of firms to respond to a cost shock.

The combined results of the paper suggest that our recursive algorithm can easily 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 highly complex settings.[2] We have shown this in the context of Monte Carlos simulations of auctions and nonlinear pricing, and using data on procurement auctions and cereals. 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, competition with menu design, optimal principal-agent contracts, 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, including unstable equilibria that are usually very hard to find. In conclusion, we imagine that the recursive method can be useful in a broad range of scenarios.

The rest of the paper is organized as follows. Section 2 describes our computational method. Section 3 details the results of our Monte Carlo study. Section 4 contains the application to highway procurement auctions. Section 5 contains the application to the ready-to-eat breakfast cereal market. Finally, Section 6 concludes.

# 2 Computational Method

In this section we present a general single-agent optimization problem then describe the conditions on this problem that make it possible to solve it using our approach, and introduce the recursive single-agent algorithm. Then we extend this approach to two methods for solving for equilibrium: the recursive full equilibrium algorithm, and the recursive iterated algorithm.

## 2.1 A Single-agent Problem

Consider a single agent who chooses an action mapping $x(t)$ for each state $t \in T$, where $T$ is an ordered set in which $\underline{t}$ and $\bar{t}$ are, respectively, the smallest and largest element, and $x : T \to Y$ belongs to a functional space $\mathcal{X}$. For instance, $x$ could be a price, and $t$ a player type or the quantity of a good. We let $t$ conditional on the function $x$ be distributed according to distribution function $F(t|x)$. We suppose that the agent is subject to a set of constraints given by $\mathbf{R}(\cdot)$. Let $\Pi(x(t), t)$ denote the ex-post payoff. The agent then solves the following problem, which we denote a *pricing game*:

$$\begin{aligned} \underset{x(t) \in \mathcal{X}}{\text{maximize}} \quad & \int_T \Pi(x(t), t)\, dF(t|x) \\ \text{subject to} \quad & \mathbf{R}(x(t), t; x(t'), t') \geq 0, \text{ for all } t, t' \in T. \end{aligned} \tag{1}$$

Many standard problems in economics can be described under this setup, including problems in the economics of optimal taxation, public good provision, nonlinear pricing, imperfect competition in differentiated industries, regulation with information asymmetries, government procurement, corporate finance, and auctions. In these problems $R$

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

can capture many things including incentive compatibility or individual rationality constraints, or monotonicity restrictions.

We next give some examples. In particular, we show how to write a first-price single-unit auction and a non-linear pricing problem as pricing games. Later on, we will focus on these two examples in our Monte Carlo simulations and applications. Many related problems can also be written in this form, for instance multi-unit auctions or contracting problems.

**Example 1** (Single-unit, 2-bidders, first-price auction: strategy space)**.** *We are interested in solving the single-agent decision problem of bidder 1. Let $x(t)$ specify bidder 1's bid following any realization of their value $t$, which has distribution $G$. Bidder 2 bids $b \sim H$. Bidder 1's profit function conditional on value $t$ and bid $x(t)$ is $\Pi(x(t), t) = H(x(t)) \times (t - x(t))$ with $F(t|x) = G(t)$ the distribution of values. There are no relevant constraints.*

The same problem can instead be formulated in the inverse-strategy space. To simplify, we assume that bid functions are strictly monotone.

**Example 2** (Single-unit, 2-bidders, first-price auction: inverse strategy space)**.** *Suppose two bidders with privately observed types bid in a single-unit first-price auction. In inverse strategy space, $t$ denotes the price, and $x(t)$ the smallest type of bidder 1 that is willing to bid at or above $t$. Bidder 2 bids $b \sim H$. Bidder 1's profit function conditional on bid $t$ and value $x(t)$ is $\Pi(x(t), t) = H(t) \times (x(t) - t)$, with $F(t|x) = G(x(t))$ the distribution of bidder 1's bids. To assure monotonicity, we impose the constraint*

$$\mathbf{R}(x(t), t; x(t'), t') = x(t) - x(t') \geq 0 \text{ for all } t \geq t'.$$

**Example 3** (Non-linear pricing)**.** *Suppose there is a set of consumers with type $t \sim G$ and utility function $u(p, q, t)$. Let $x(t) = (p(t), q(t))$ be a price-quantity pair offered by the firm. The firm's profit function is*

$$\Pi(x(t), t) = p(t) - cq(t),$$

*with $F(t|x) = G(t)$. The set of incentive compatibility and participation constraints is given by*

$$\mathbf{R}(x(t), t; x(t'), t') = \begin{cases} u(p(t), q(t), t) - u(p(t'), q(t'), t) \geq 0 & \text{for all } t, t' \in T; \\ u(p(t), q(t), t) \geq 0 & \text{for all } t \in T. \end{cases}$$

For any such problem we can take a discrete approximation of equilibrium strategies. This is a special case of a polynomial approximation. Approximation via polynomial basis functions is a well studied problem in numerical analysis and the choice of basis function families can impact computation and convergence (e.g., Judd (1998)).

Let us construct the following discrete version of the pricing game described in (1). First we define the following discrete equivalents of the primitives of the problem.

- Let $\hat{T}$ denote a partition of $T$ into $N$ elements, such that $\hat{T} = \{t_1, .., t_N\}$.
- Let $\hat{Y}$ denote a partition of $Y$ into $K$ elements, $\hat{Y} = \{y_1, ..., y_K\}$.

- Let $\hat{x} : \hat{T} \to \hat{Y}$ denote the corresponding discrete decision function with $\hat{\mathcal{X}}$ the corresponding functional space. Let $\hat{x}_n = \hat{x}(t_n)$ and $\hat{\mathbf{x}} = (\hat{x}_1, ..., \hat{x}_N)$.
- For $n \in \{1, ..., N\}$, let $\mu_n(\hat{\mathbf{x}})$ denote the mass of region $n$ and let $t_n$ be some point inside the region.

We then define the following discrete equivalent of the pricing game, which we denote a *discrete pricing game*:

$$\begin{aligned} \underset{\hat{x}_n \in \hat{\mathcal{X}}}{\text{maximize}} \quad & \sum_n \Pi(\hat{x}_n, t_n)\mu_n(\hat{\mathbf{x}}) \\ \text{subject to} \quad & \mathbf{R}(\hat{x}_n, t_n; \hat{x}_{n'}, t_{n'}) \geq 0, \text{ for all } n, n'. \end{aligned} \tag{2}$$

If the problem is sufficiently regular, the solution of the discretized problem in (2) is close to the solution of the original problem in (1). For example, take the single-agent auction problem and suppose that there are no constraints, so that $\mathbf{R}$ is a zero function, and $\hat{T}$ is an evenly spaced partition of $[\underline{t}, \overline{t}]$, and the points $t_n$ the midpoint of each interval. Suppose the function $\Pi(\cdot)$ has bounded second derivatives, and $\hat{Y}$ is an evenly spaced grid with intervals $\Delta_K$. Then the approximation error, as measured by the distance between (1) and (2) evaluated at their respective solutions, is bounded by $\frac{\Delta_K}{N^2} + \frac{d}{N^2}$ for constant $d$. In the applications below, we choose the grid in $\Delta_K$ to be much finer than the grid in $N$, and the errors from the first term are small. The approximation error therefore is $O(\frac{1}{N^2})$. This approximation is analogous to the midpoint rule for numerical integration. Rather than using step functions, an alternative would be to take a linear approximation within each interval. This would be analogous to the trapezoidal rule for numerical integration, and would reduce the constant term $d$, but would maintain the same order in $N$.

## 2.2 Recursive Algorithm for Single-agent Problem

We now define a recursive algorithm and show that it provides a full solution to the discrete pricing game of the previous section if it satisfies the following definition of local constraints.

**Definition 1.** *We say that a discrete pricing game has **local constraints** if for all steps $1 \leq n < N$, we can write the step mass as a function of the action at steps $n$ and $n+1$,*

$$\mu_n(\hat{\mathbf{x}}) = m_n(\hat{x}_n, \hat{x}_{n+1}),$$

*with $\mu_N(\hat{\mathbf{x}}) = m_N(\hat{x}_N)$, and that all constraints are local, either upward or downward, such that*

$$\mathbf{R}(\hat{x}_n, t_n; \hat{x}_{n+1}, t_{n+1}) \geq 0 \text{ for all } n < N \tag{3}$$

*imply $\mathbf{R}(\hat{x}_n, t_n; \hat{x}_{n'}, t_{n'}) \geq 0$ for all $n, n'$.*[3]

[3]Observe that the local constraints include constraints that depend only on $t_n$ and $\hat{x}_n$. Notice also that we could allow $\mu_n$ to depend on $x_{n-1}$ additively, but as this is not required in our applications, we maintain the simple formulation for expositional purposes.

We now illustrate how the algorithm works in a pricing game with local constraints. We will solve the problem by moving sequentially over $\hat{T}$, from the first to the last element. At each step $n = 1, ..., N$ in the sequence, and for each possible choice at the next step, $\hat{x}_{n+1}$, we solve for the action $\hat{x}_n$ that maximizes $\Pi$ over $t_1, ..., t_n$ given the optimal actions at points $n' \leq n - 1$ which are induced by our choice of $\hat{x}_n$. Thus, at each $n$, we have to calculate $K$ optimal choices, one for each of the possible actions $\hat{x}_{n+1}$ at the next step, where each of these optimal actions will take into account the optimal actions at previous steps $n' \leq n - 1$. Therefore, the algorithm can be efficiently computed when the optimal action at any step $n$ depends only on the optimal actions at the next and at the previous step.

The algorithm reduces the number of calculations required to obtain a global solution of the maximization problem to the following: at each step $n = 1, ..., N$, for each possible action at the next step $k = 1, ..., K$, there are at most $K$ actions available. Hence, we require at most $N \times K^2$ calculations to obtain the matrix of optimal actions at each step conditional on the optimal action at the next step. Furthermore, we need $N$ calculations to derive the sequence of optimal choices from the matrix of conditional optimal actions, and thus we need $N \times K^2 + N$ calculations in total. In comparison, the only alternative known to obtain a global solution, an exhaustive grid search, would require $K^N$ calculations to calculate the value of each action sequence, and then a calculation to choose the maximum of these. Hence, for $N = K = 10$ the recursive algorithm requires $10 \times 10^2 + 10 = 1,010$ calculations whereas the grid search requires $10^{10}$ calculations. Notice that both our algorithm and the exhaustive grid search can be simplified somewhat when the problem includes monotonicity restrictions on $x$.

We next describe the recursive algorithm in detail.

**Algorithm 1** (Recursive Single-agent Algorithm)**.** *Let $y$ denote the action at $t_n$ and let $z$ denote the action at $t_{n+1}$. At each $n < N$, let $C_n(z)$ be the set of $y \in \hat{Y}$ that satisfy the local constraints*

$$\mathbf{R}(y, t_n; z, t_{n+1}) \geq 0. \tag{4}$$

*If $C_n(z)$ is empty for some $z$ and $n$, we let $\max_{y \in C_n(z)}\{\cdot\} = -\infty$. The algorithm then consists of the following steps:*

- *Step 1. At $n = 1$. For each $z \in \hat{Y}$, define the equation*

$$V_1(z) \equiv \max_{y \in C_1(z)} \{\Pi(y, t_1) m_1(y, z)\}.$$

- *Step 2. At $1 < n < N$: For each $z \in \hat{Y}$, define the Bellman equation*

$$V_n(z) \equiv \max_{y \in C_n(z)} \{V_{n-1}(y) + \Pi(y, t_n) m_n(y, z)\}. \tag{5}$$

  *At $n = N$ we replace $m_n(y, z)$ and $C_n(z)$ by $m_N(y)$ and $\hat{Y}$.*

- *Step 3. For each $n$, let $\bar{x}_n(z)$ be a solution (argmax) to the right-hand side of $V_n(z)$. Calculate solution $\mathbf{x}^* = (x_1^*, ..., x_N^*)$ recursively by applying*

$$x_n^* = \bar{x}_n(x_{n+1}^*) \text{ for all } n < N.$$

Notice that we maximize at step $n$ conditional on the action at step $n+1$, and therefore we can incorporate the local constraints in both directions by making the maximization subject to the step $n$ upward constraints and the step $n+1$ downward constraints. The step $n$ downward constraint is has already been incorporated in $V_{n-1}(\cdot)$.

Multiple solutions can be captured by allowing $\bar{x}_n(z)$ to be set valued. In this case, in the final step of the algorithm, the different solutions will be found by following the different 'paths' through the conditional step solutions $\bar{x}_n(z)$. Given a discrete pricing game with local constraints, denote by $\mathbf{X}^\dagger$ the set of solutions to the maximization problem (2), and denote by $\mathbf{X}^*$ the set of solutions to Algorithm 1. We say that a discrete pricing game with local constraints is *recursively solvable* if $\mathbf{X}^\dagger = \mathbf{X}^*$.

We then show the following result.

**Proposition 1.** *Any discrete pricing game with local constraints is recursively solvable.*

The proof is in Appendix A.

One might be concerned that because we only verify the constraints at the set of grid-points, this relaxed problem has additional feasible solutions that differ substantially from the true solution and allow for higher profits. This could occur because the relaxed constraints may allow a bidder to take a step that is feasible only under a sub-optimal path through the intermediate points. Since we choose a grid that is much finer in the dimension $t$ than $x$, the restrictions from the optimal path between $t$ and $t+1$ will usually not refine the set of possible $x(t)$ by more than the size of the grid-spacing in $x$-space. Thus while the set of points satisfying the relaxed constraint expands, it will not contain additional points that belong to the grid used in approximation, and the solution is unchanged.[4]

Mechanism design problems in which local incentive constraints are sufficient for full incentive compatibility satisfy the conditions of the proposition, and are therefore recursively solvable. This includes any standard auction (single or multiple units) or settings that satisfy the conditions of Carroll (2012), such as our nonlinear pricing example.[5] It is important to note that our approach embeds the two sides of the incentive compatibility condition rather than the one side alone. In parametric utility approaches or approaches following Matthews and Moore (1987), it is often assumed that only downward incentive compatibility constraints bind, but we do not require this.[6]

## 2.3 Solving for Equilibrium

We now wish to extend the single-agent tools developed so far to solving for equilibrium in games. Our approach fits within the notion of Constrained Strategic Equilibrium of Armantier, Florens, and Richard (2008), implying that as the approximation errors in

[4]Alternatively, additional restrictions on the slope and concavity of the profit functions could be used to place a formal upper bound on these errors.

[5]In particular, Carroll (2012) shows that on a convex cardinal type space $T$, any set of local incentive constraints is sufficient.

[6]When $t$ is unidimensional, the real line provides a natural ordering of the incentive compatibility constraints. In two dimensions, this problem is harder, and a curse of dimensionality may arise.

each agent's action function get small, the solution will converge to the true equilibrium.[7] For small grids, this can play an important role, but once a reasonable grid density is reached, it appears to have only a small impact on the solution.[8] However, our approach to solving the equilibrium differs in important ways. In contrast to Armantier et al. (2008), where parameters are adjusted at each iteration towards a local maximum, we either solve a full globally optimal strategy that respects the equilibrium constraints, or solve a global best response of each bidder which we use to update strategies sequentially. This helps to avoid making incorrect adjustments towards local optima.

We next detail the model. Suppose there are $i = 1, ..., I$ agents who choose action mappings $\mathbf{x}(t) = (x_1(t), ..., x_I(t))$. Let $\mathbf{x}_{-i}$ denote the set of strategies for all agents except $i$. Each agent $i$ solves

$$\begin{array}{ll} \underset{x_i(t) \in \mathcal{X}}{\text{maximize}} & \displaystyle\int_T \Pi_i(x_i(t), \mathbf{x}_{-i}(t), t) F_i(t|x_i, \mathbf{x}_{-i}) \\ \text{subject to} & \mathbf{R}_i(\mathbf{x}(t), t; \mathbf{x}(t'), t') \geq 0, \text{ for all } t, t' \in T. \end{array} \tag{6}$$

Notice that we allow $\Pi_i$, $F_i$ and $\mathbf{R}_i$ to depend on the actions of other agents as well. An *equilibrium* in this setting is a mapping $\mathbf{x}^*$ such that each $x_i^*$ solves the agent's maximization problem conditional on $\mathbf{x}^*_{-i}$. The corresponding discrete problem with local constraints defines a mapping $\hat{\mathbf{x}}_n = (\hat{x}_{1,n}, ..., \hat{x}_{I,n})$ with $\hat{\mathbf{x}}_{-i,n}$ being the strategies for all agents except $i$. Each agent $i$ then solves

$$\begin{array}{ll} \underset{\hat{x}_{i,n} \in \hat{\mathcal{X}}}{\text{maximize}} & \displaystyle\sum_n \Pi_i(\hat{x}_{i,n}, \hat{\mathbf{x}}_{-i,n}, t_n) m_{i,n}(\hat{x}_{n,i}, \hat{\mathbf{x}}_{n,-i}, \hat{\mathbf{x}}_{n+1}) \\ \text{subject to} & \mathbf{R}_i(\hat{\mathbf{x}}_n, t_n; \hat{\mathbf{x}}_{n+1}, t_{n+1}) \geq 0, \text{ for all } n < N, \end{array} \tag{7}$$

where $m_{i,n}(\hat{x}_{n,i}, \hat{\mathbf{x}}_{n,-i}, \hat{\mathbf{x}}_{n+1})$ at $n = N$ is given by $m_{i,N}(\hat{x}_{N,i}, \hat{\mathbf{x}}_{N,-i})$. The equilibrium is defined equivalently to the continuous case.

A variety of formulations for solving the game are possible and may be useful in different contexts. We will use two methods: full equilibrium, and iterated equilibrium. We next describe each in turn.

**Full equilibrium.** The direct approach to the above problem, is to solve for the optimal path jointly with the best response. At step $n$, we perform two steps, first solve an optimization problem, then check a fixed point condition. The algorithm proceeds as Algorithm 1 with the following adjustments.

**Algorithm 2** (Recursive Full Equilibrium Algorithm)**.** *Let $y$ and $\mathbf{y}$ denote, respectively, the individual and joint actions at $t_n$, and let $z$ and $\mathbf{z}$ denote, respectively, the individual and joint actions at $t_{n+1}$. Assume that both the mutual and individual best responses of the agents at each $n$ are unique. At each $n < N$, let $C_n(\mathbf{z}, \mathbf{y}_{-i})$ be the set of $y \in \hat{Y}$ that satisfy the local constraints*

$$\mathbf{R}_i(y, \mathbf{y}_{-i}, t_n; \mathbf{z}, t_{n+1}) \geq 0. \tag{8}$$

[7] In the auction game, Reny and Zamir (2004) show that a monotone equilibrium exists when bidders are restricted to a finite set of bids, and that the limit of these equilibria as the set of bids increases, is an equilibrium in the continuous auction game.

[8] Since errors in our approximation converge quickly to zero, convergence turns out to be a minimal concern in practice. See Section 3.

*If $C_n(\mathbf{z}, \mathbf{y}_{-i})$ is empty for some $\mathbf{y}_{-i}$, $\mathbf{z}$ and $n$, we let $\max_{y \in C_n(\mathbf{z},\mathbf{y}_{-i})}\{\cdot\} = -\infty$. The algorithm then consists of the following steps:*

- *Step 1. At $n = 1$. For each $i$, for each $\mathbf{z} \in \hat{Y}^I$ and each vector of other agents' actions, $\mathbf{y}_{-i} \in \hat{Y}^{I-1}$, agent $i$ solves*

$$\max_{y \in C_1(\mathbf{z},\mathbf{y}_{-i})} \{\Pi_i(y, \mathbf{y}_{-i}, t_1) m_{i,1}(y, \mathbf{y}_{-i}, \mathbf{z})\}.$$

  *Let $\bar{\mathbf{x}}_1(\mathbf{z})$ denote the (unique) mutual best response. Define the value function*

$$V_{i,1}(\mathbf{z}) \equiv \Pi_i(\bar{\mathbf{x}}_1(\mathbf{z}), t_1) m_{i,1}(\bar{\mathbf{x}}_1(\mathbf{z}), \mathbf{z}).$$

- *Step 2. At $1 < n < N$: For each $\mathbf{z} \in \hat{Y}^I$ and each vector of other agents' actions, $\mathbf{y}_{-i} \in \hat{Y}^{I-1}$, agent $i$ solves*

$$\max_{y \in C_n(\mathbf{z},\mathbf{y}_{-i})} \{V_{i,n-1}(y, \mathbf{y}_{-i}) + \Pi_i(y, \mathbf{y}_{-i}, t_n) m_{i,n}(y, \mathbf{y}_{-i}, \mathbf{z})\}.$$

  *Let $\bar{\mathbf{x}}_n(\mathbf{z})$ denote the (unique) mutual best response. Then define the value function*

$$V_{i,n}(\mathbf{z}) \equiv V_{i,n-1}(\bar{\mathbf{x}}_n(\mathbf{z})) + \Pi_i(\bar{\mathbf{x}}_n(\mathbf{z}), t_n) m_{i,n}(\bar{\mathbf{x}}_n(\mathbf{z}), \mathbf{z}). \tag{9}$$

  *At $n = N$ we replace $m_{i,n}(y, \mathbf{y}_{-i}, \mathbf{z})$ and $C_n(\mathbf{z}, \mathbf{y}_{-i})$ by $m_N(y, \mathbf{y}_{-i})$ and $\hat{Y}$.*

- *Step 3. Calculate solution $\mathbf{x}^* = (\mathbf{x}_1^*, ..., \mathbf{x}_N^*)$ recursively by applying $\mathbf{x}_n^* = \bar{\mathbf{x}}_n(\mathbf{x}_{n+1}^*)$ for all $n < N$.*

In this formulation, at each step the state is all possible own and opponent actions. This means that the state space expands quickly in the number of bidders, leading to a curse of dimensionality when the number of bidders is large.[9] However, it is instructive to compare the severity of the curse. The recursive full equilibrium algorithm requires $N \times (K^I)^2 + N \times I$ calculations. In comparison, if we were to find the full solution using a grid search, which as before is the only known alternative to obtain a full global solution, we would require $(K^N)^I$ calculations. With $N = K = 10$ and $I = 2$, this corresponds to $10 \times (10^2)^2 + 10 \times 2 = 10^5 + 20$ versus $(10^{10})^2 = 10^{20}$ calculations.

Equilibrium multiplicity can be dealt with in the following manner. If the mutual best response is not unique at some $n$, we can let the algorithm 'fork out', which will imply that we have to calculate the algorithm an extra time for $n' > n$. Each time the algorithm forks out, we thus create a new matrix of conditional mutual best responses, and all equilibria can then be found as a path through on of these matrices.

**Iterated equilibrium.** The expansion of the state space required to calculate the full equilibrium can be prohibitively costly in terms of computational time. Therefore, we will mainly adopt an iterated approach that avoids the expansion of the state-space and appears to both require little computation time and to converge well in practice. In this approach, we cycle through the agents, and for each one solve the best response against

[9] If we restrict ourselves to searching for symmetric equilibria, we can replace bidders by groups of bidders who have the same primitives $\Pi_i$ and $m_{i,n}$, which reduces the dimensionality curse.

the current behavior of opponents using our recursive algorithm. In many examples, competition is captured by the individual rationality constraint. This approach allows the state space to remain the same size as in the individual optimization problem, and therefore may lead to computational savings.

**Algorithm 3** (Recursive Iterated Equilibrium Algorithm)**.** *Choose a tolerance level $\epsilon > 0$ and a distance measure $D(\cdot, \cdot)$ for convergence.*

- *Step 1. Specify an initial strategy $\mathbf{x}^{(0)}$.*
- *Step 2. Iteration $s \geq 1$. For all $i$ and $n$, extract $\Pi_i$, $m_{i,n}$ and $\mathbf{R}_i$ conditional on $\mathbf{x}^{(s-1)}$:*

$$\begin{aligned}\Pi_i^{(s)}(x_{i,n}, t) &\equiv \Pi_i(x_{i,n}, \mathbf{x}_{-i,n}^{(s-1)}, t_n);\\ m_{i,n}^{(s)}(x_{i,n-1}, x_{i,n}, x_{i,n+1}) &\equiv m_{i,n}(x_{i,n}, \mathbf{x}_{-i,n}^{(s-1)}, x_{i,n+1}, \mathbf{x}_{-i,n+1}^{(s-1)});\\ \mathbf{R}_i^{(s)}(x_{i,n}, t_n; x_{i,n'}, t_{n+1}) &\equiv \mathbf{R}_i(x_{i,n}, \mathbf{x}_{-i}^{(s-1)}, t_n, x_{i,n+1}, \mathbf{x}_{-i}^{(s-1)}, t_{n+1}).\end{aligned}$$

- *Step 3. For each $i$, solve the single-agent problem conditional on $\Pi_i^{(s)}$, $m_{i,n}^{(s)}$ and the constraints $\mathbf{R}_i^{(s)}$ using Algorithm 1 and denote the solution of all $i$ at all $n$ by $\mathbf{x}^{(s)}$.*
- *Step 4. Stop algorithm if $D(\mathbf{x}^{(s)}, \mathbf{x}^{(s-1)}) < \epsilon$; if not, go back to step 2 and start iteration $s + 1$.*

Depending on the application, we sometimes apply a dampening parameter to the update of the functions $\Pi_i^{(s)}$, $m_{i,n}^{(s)}$ and $\mathbf{R}_i^{(s)}$ to improve convergence.

This iterative procedure performs well in the simulations we have considered in the sense that it converges quickly and with small errors. However, as with many approaches based on iterating best responses, there is no guarantee that this approach will converge, or that it will converge to the correct solution.[10] Other alternatives to the iterated approach are also possible, for example, in some settings it may be possible to reduce the state-space to a sufficient statistic.[11]

The iterated equilibrium algorithm cannot capture equilibrium multiplicity in the same manner as the previous two algorithms. Standard methods such as using different initial strategies can be applied, and we conjecture that these will work reasonably well with stable equilibria, but they will rarely be able to capture unstable equilibria. If it is suspected that the model has unstable equilibria or if equilibrium multiplicity is a priority, the full equilibrium algorithm should be used. Homotopy methods similar to Borkovsky, Doraszelski, and Kryukov (2010) may also be useful to find other equilibria, with the solution from the iterated algorithm serving as a starting point.

[10] Although we have not seen cases where this is an issue in practice, if this were suspected to be an issue in some example, a natural approach would be to obtain good starting values for the iterated approach by first using the full joint solution approach with coarse grids.

[11] One natural proposal for such a sufficient statistic in the auction context is the probability that all opposing bids lie below the current price level. While this is a sufficient statistic in the individual bidder optimization problem, it is not sufficient for best response verification in the equilibrium solution. However, alternative methods of verification or alternative statistics may be feasible.

# 3 Monte Carlo

## 3.1 Asymmetric Auction

We first apply our methodology 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. All existing numerical approaches for finding auction equilibria are based on the solution of the (system) of differential equations that characterize optimal bidding, but in asymmetric settings, the solution often becomes complex, especially if the supports of the underlying value distributions differ.[12] We first give a brief overview of previous approaches, and then discuss our approach.

A common approach for solving the system of differential equations in asymmetric auctions is the *shooting algorithm* (Marshall, Meurer, Richard, and Stromquist, 1994; Bajari, 2001; Gayle and Richard, 2008). This method solves the problem as if it were 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. There are two sources of instability. First, the shooting algorithm is extremely sensitive to the choice of initial condition by construction.[13] Second, additional errors may accumulate along the path. In the case where the initial bid gives too low surplus to bidders the error is interpretable, the algorithm converges to zero surplus too soon and it is as if the bidders were participating in an auction with a reserve price at the point where they reach zero surplus. To correct this problem they transform the upper boundary condition so that the transformed system has a fixed domain, and then solve this as a fixed point or via Newton's method. However, unlike our method which adapts easily to variation in the support of values, their approach requires a common support. In the subset of cases where their approach can be applied, it achieves similar computational time to the polynomial approximation and, in turn, to our approach.

Another common 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).[14] This approach typically uses Chebyshev polynomials as the basis for the projection, and solves for the parameters that minimize the errors in the differential equation subject to the boundary constraints. However, 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 also sensitive to initial values. Furthermore, since the projection method provides an approximation, it may be difficult to implement shape constraints. One approach is to impose further constraints such as rationality and monotonicity, as in Hubbard et al. (2013).

Separate from these issues are the well-known problems with numerical stability of

[12]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).

[13]Consider a symmetric auction solved via backward shooting starting $\epsilon$ 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.

[14]Notice that Bajari (2001) uses both the shooting algorithm and projection methods.

the numerical methods commonly used for solving the system of differential equations such as stiffness (sensitivity to step size), estimation of higher-order derivatives to apply Taylor approximations and the lack of a common upper bound when there are more than 3 bidders.[15] The latter issue, which often leads to bid bifurcation, is addressed in Hubbard and Kirkegaard (2015), but their method cannot be easily extended to cases with more than two bidder groups, or to allow the supports to have both different lower and upper bounds. Dharanan and Ellis (2024) propose a method for solving the differential equations in these more complex cases, but this requires an approximate solution method based on extending the support of the value distributions, together with specialized numerical tools for stiff differential equations and is computationally intensive.

Our approach differs fundamentally from the above approaches in that we are not solving a system of differential equations locally along a path. Our approach looks globally, across our entire admissible set of functions, to find the function that maximizes profits. Since our method is based on selecting the optimal action at each step along the path, and does not solve a differential equation it will not suffer from the aforementoioned stability issues. Our approach is similar to the *projection methods*, but our choice of step-functions as the basis for the approximation allows for global optimization, via the recursive representation. This allows us to obtain a global solution quickly, and since we do not rely on the differential equation representation, we require less restrictive conditions, and problem-specific constraints. Below, we first outline our approach, and then we analyze a set of examples that show how our method compares to the aforementioned, and in particular, how it deals with the issues that they face.

**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 and formulate the problem in discrete space with local constraints as in (7). First, we specify a price grid $\hat{T}$ and a type grid $\hat{X}$ such that $\hat{x}_{i,n} \in \hat{X}$ is the smallest value on $\hat{X}$ for which bidder $i$ bids $t_n \in \hat{T}$. Hence, bidder $i$'s expected profit conditional on bidding price $t$ is

$$\Pi_i(\hat{x}_n, t_n) = \left(\prod_{j \neq i} G_j(\hat{x}_{j,n})\right) \times (\mathbb{E}[\hat{x}_{i,n} \leq v_i < \hat{x}_{i,n+1}] - t_n),$$

with $m_{i,n} = G_i(\hat{x}_{i,n+1}) - G_i(\hat{x}_{i,n})$ and the constraints $\mathbf{R}_i(\hat{\mathbf{x}}_n, t_n; \hat{\mathbf{x}}_{n+1}, t_{n+1}) = \hat{x}_{i,n+1} - \hat{x}_{i,n} \geq 0$ for all $i$ and $n < N$.[16]

We can then implement Algorithms 2 and 3. In the latter, we let $m_{i,n}$ be updated with a dampening weight. We make win probabilities smooth at each iteration to eliminate discontinuities due to the discreteness of opposing bidders' bid strategies.[17] The baseline results used a tolerance of $\epsilon = 0.01$.

[15] These issues are discussed in Hubbard and Paarsch (2014) and they also point out that boundary conditions may need adjustment if type supports differ with more than 2 bidders, the subject of Hubbard and Kirkegaard (2015).

[16] Notice that $\Pi_i(\hat{x}_n, t_n)$ also depends on $\hat{x}_{i,n+1}$: this does not affect the properties of the algorithm, as the local constraints assumption could merely be extended to include this. However, we prefer to maintain the notation simple.

[17] 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.

**Simulations.** To assess our algorithm, we use the following set of examples. These examples allow us to evaluate the performance of our approach by comparing our solution to the solution obtained using alternative numerical methods. For all cases, the existence and characterization of the equilibrium follow from Lebrun (2006).

- *Example 1.* 2 bidders, with values distributed uniform [0,1], i.e., $G_1 = G_2 = U(0, 1)$.
- *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 $G_1 = U(0, \frac{4}{5})$, and $G_2 = U(0, \frac{4}{3})$
- *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, $N_x$ refers to the number of grid points in the grid for the action (the smallest valuation bidding at step), $x$, and $N_t$ is the number of grid points in the grid for the type (the price), $t$. We next discuss the examples in turn.

Example 1 can be solved analytically, and is also used as a benchmark to evaluate the performance of the shooting method, and the polynomial collocation approach in Hubbard and Paarsch (2014).[18] In this simple setting, the shooting method, polynomial approximations and our solution method all obtain accurate solutions. The polynomial method, however can be unstable as the number of coefficients used in the approximation gets large (around 30). While the solution at $M = 30$ polynomial terms results in vastly different profits, with a slight increase in the number of coefficients, the solution at $M = 32$ approximates profits well. In contrast, our solution appears to be stable with respect to a wide range of choices for the number of points in both the grid of prices and value cutoffs. Our solution requires a similar computation time to the polynomial approach, and is much faster than the shooting algorithm. Both the computation time and the accuracy of the solution from the polynomial approach degrade much more quickly than our step-function approach as the number of strategies to solve increases, (e.g. expanding the number of groups of asymmetric bidders). With 3 or 4 different strategy functions, the polynomial approach fails to converge to the global payoff maximizing functions, while our approach continues to converge. In addition, our approach requires only small increases in computational time.

[18]This corresponds to Example 1 of Hubbard and Paarsch (2014). They also consider an iterative method, proposed by Fibich and Gavish (2011), which they demonstrate has similar performance to the polynomial approach. Since it has similar performance, but cannot be used in the more complex examples such as 4 and 5, we do not further consider this method here.

Table 1: Asymmetric Auction Iterated Solution – Monte Carlo

| | Profit | | Compute |
|---|---|---|---|
| | Bidder 1 | Bidder 2 | Time (s) |
| *Example 1* | | | |
| Analytic | 0.1667 | 0.1667 | - |
| Shooting | 0.1666 | 0.1666 | 120.71 |
| Polynomial ($M = 10$) | 0.1664 | 0.1664 | 0.05 |
| Polynomial ($M = 20$) | 0.1664 | 0.1664 | 0.56 |
| Polynomial ($M = 30$) | $-0.1200$ | 0.0400 | 2.19 |
| Recursive ($N_x = 5, N_t = 100$) | 0.1789 | 0.1789 | 0.12 |
| Recursive ($N_x = 10, N_t = 100$) | 0.1657 | 0.1658 | 0.45 |
| Recursive ($N_x = 20, N_t = 200$) | 0.1658 | 0.1658 | 1.96 |
| Recursive ($N_x = 30, N_t = 200$) | 0.1680 | 0.1680 | 1.07 |
| *Example 2* | | | |
| Shooting | 0.1015 | 0.2276 | 30.33 |
| Polynomial ($M = 10$) | 0.1010 | 0.2270 | 0.19 |
| Polynomial ($M = 20$) | 0.1010 | 0.2270 | 0.37 |
| Polynomial ($M = 30$) | 0.0880 | 0.2150 | 1.778 |
| Recursive ($N_x = 5, N_t = 100$) | 0.1202 | 0.2813 | 0.23 |
| Recursive ($N_x = 10, N_t = 100$) | 0.1140 | 0.2443 | 0.92 |
| Recursive ($N_x = 20, N_t = 200$) | 0.1104 | 0.2344 | 5.68 |
| Recursive ($N_x = 30, N_t = 200$) | 0.1083 | 0.2295 | 3.59 |
| *Example 3* | | | |
| Analytic | 0.0980 | 0.3110 | - |
| Recursive ($N_x = 5, N_t = 100$) | 0.0935 | 0.3138 | 0.15 |
| Recursive ($N_x = 10, N_t = 100$) | 0.0842 | 0.2970 | 0.61 |
| Recursive ($N_x = 20, N_t = 200$) | 0.0890 | 0.3038 | 2.76 |
| Recursive ($N_x = 30, N_t = 200$) | 0.0891 | 0.3046 | 1.52 |
| *Example 4* | | | |
| Polynomial ($M = 10$) | 0.0618 | 0.0291 | 0.21 |
| Polynomial ($M = 20$) | 0.0575 | 0.0170 | 1.2 |
| Polynomial ($M = 30$) | 0.0599 | 0.0193 | 1.41 |
| Recursive ($N_x = 5, N_t = 100$) | 0.0658 | 0.0112 | 0.20 |
| Recursive ($N_x = 10, N_t = 100$) | 0.0553 | 0.0113 | 0.72 |
| Recursive ($N_x = 20, N_t = 200$) | 0.0529 | 0.0105 | 3.62 |
| Recursive ($N_x = 30, N_t = 200$) | 0.0547 | 0.0109 | 2.09 |
| *Example 5* | | | |
| Recursive ($N_x = 5, N_t = 100$) | 1.0120 | 0.6155 | 0.26 |
| Recursive ($N_x = 10, N_t = 100$) | 0.8804 | 0.5007 | 1.16 |
| Recursive ($N_x = 20, N_t = 200$) | 0.8819 | 0.5062 | 6.14 |
| Recursive ($N_x = 30, N_t = 200$) | 0.8750 | 0.4897 | 4.71 |

Note: *Shooting* and *Polynomial*: use the Hubbard and Paarsch (2014) code, which leverages the *SNOPT* package. For the polynomial, $M$ indicates the number of Chebyshev polynomial terms (and the number of coefficients) used. *Recursive*: the Recursive Iterated Equilibrium Algorithm. *Analytic*: analytic solution (hence compute time is not relevant). The polynomial approach in Example 4 differs from that of Examples 1 and 2. The solution is obtained using a standard solver in Matlab, *fmincon*, rather than the *SNOPT* package, and beyond the common subset of the support an analytic solution is used, following Hubbard and Kirkegaard (2015).

Figure 2: Auction Initial Value Sensitivity

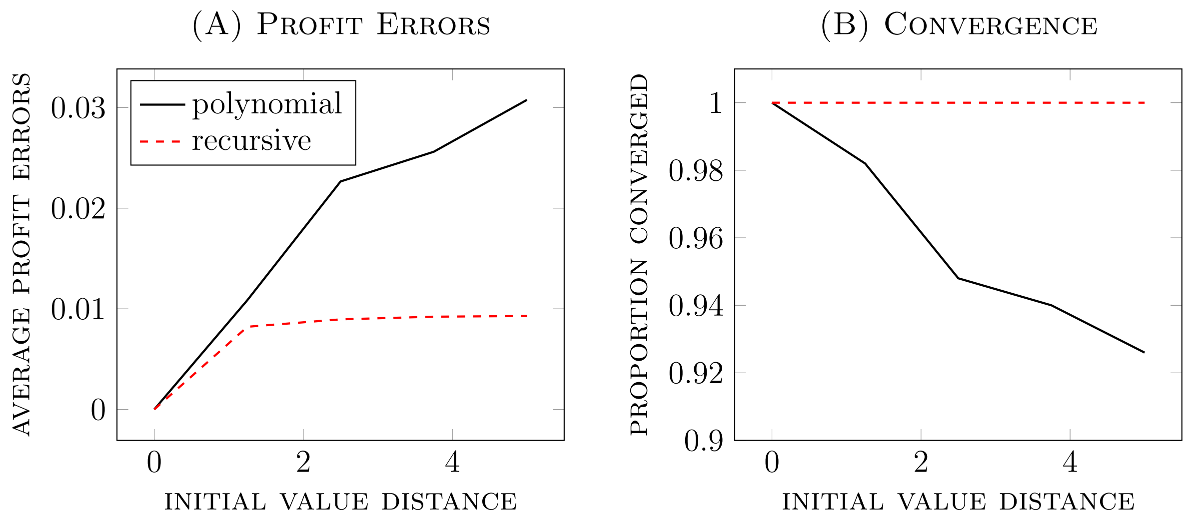



Figure 2 illustrates the sensitivity of the polynomial approximation approach to initial values with a chebyshev polynomial with $M = 20$. Taking random perturbations of the initial values, $\beta_0 = d \times (\frac{r}{\sum r})$, where $\beta_0$ are the parameters describing the true solution, $d$ is a constant chosen by the researcher and $r$ is a vector of uniform random draws. Computing the equilibrium for each one, we find the average errors in profits expand more quickly for the polynomial approach than our recursive solution, and that the polynomial approach quickly begins to return solutions for many starting values with profits more than 0.1 from the true solution.

Example 2 introduces asymmetry into the auction.[19] 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, with the shooting algorithm being much slower, and the quality of the polynomial approximation degrading as the number of terms included increases. 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 (2015) propose a method for numerically solving the group-symmetric 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.[20] Their method uses a polynomial approximation similar to Hubbard and Paarsch (2014), to approximate the inverse-strategy functions on the overlapping segment of the bid functions, and then solves a differential equation (which has an analytic solution) outside of the common segment, where only one bidder

[19]This corresponds to Example 2 of Hubbard and Paarsch (2014).

[20]The previous theoretical literature only had results for the cases of either two bidders, or an arbitrary number of bidders and a common support of types.

Table 2: Asymmetric Auction Full Solution – Monte Carlo

| | Profit | | Compute |
|---|---|---|---|
| | Bidder 1 | Bidder 2 | Time (s) |
| *Example 1* | | | |
| Recursive ($N_x = 5, N_t = 50, N_\pi = 50$) | 0.1493 | 0.1493 | 0.45 |
| Recursive ($N_x = 5, N_t = 50, N_\pi = 100$) | 0.1486 | 0.1486 | 0.86 |
| Recursive ($N_x = 5, N_t = 100, N_\pi = 50$) | 0.1704 | 0.1704 | 0.85 |
| Recursive ($N_x = 5, N_t = 100, N_\pi = 100$) | 0.1576 | 0.1576 | 1.62 |
| Recursive ($N_x = 5, N_t = 200, N_\pi = 200$) | 0.1635 | 0.1635 | 6.81 |
| Recursive ($N_x = 10, N_t = 50, N_\pi = 50$) | 0.1568 | 0.1568 | 1.72 |
| Recursive ($N_x = 10, N_t = 50, N_\pi = 100$) | 0.1549 | 0.1549 | 3.39 |
| Recursive ($N_x = 10, N_t = 100, N_\pi = 50$) | 0.1752 | 0.1752 | 3.47 |
| Recursive ($N_x = 10, N_t = 100, N_\pi = 100$) | 0.1610 | 0.1610 | 6.81 |
| Recursive ($N_x = 10, N_t = 200, N_\pi = 200$) | 0.1667 | 0.1667 | 29.85 |
| *Example 2* | | | |
| Recursive ($N_x = 5, N_t = 50, N_\pi = 50$) | 0.1420 | 0.2896 | 0.45 |
| Recursive ($N_x = 5, N_t = 100, N_\pi = 100$) | 0.1520 | 0.3060 | 1.52 |
| Recursive ($N_x = 10, N_t = 50, N_\pi = 50$) | 0.1354 | 0.2482 | 1.63 |
| Recursive ($N_x = 10, N_t = 100, N_\pi = 100$) | 0.1499 | 0.2689 | 6.36 |
| *Example 3* | | | |
| Recursive ($N_x = 5, N_t = 50, N_\pi = 50$) | 0.1092 | 0.3103 | 0.42 |
| Recursive ($N_x = 5, N_t = 100, N_\pi = 100$) | 0.1172 | 0.3273 | 1.50 |
| Recursive ($N_x = 10, N_t = 50, N_\pi = 50$) | 0.1016 | 0.2965 | 1.62 |
| Recursive ($N_x = 10, N_t = 100, N_\pi = 100$) | 0.1183 | 0.3220 | 6.37 |
| *Example 4* | | | |
| Recursive ($N_x = 5, N_t = 50, N_\pi = 50$) | 0.0597 | 0.0204 | 0.43 |
| Recursive ($N_x = 5, N_t = 100, N_\pi = 100$) | 0.0608 | 0.0203 | 1.51 |
| Recursive ($N_x = 10, N_t = 50, N_\pi = 50$) | 0.0708 | 0.0298 | 1.63 |
| Recursive ($N_x = 10, N_t = 100, N_\pi = 100$) | 0.0714 | 0.0285 | 6.25 |
| *Example 5* | | | |
| Recursive ($N_x = 5, N_t = 50, N_\pi = 50$) | 0.9275 | 0.6383 | 2.91 |
| Recursive ($N_x = 5, N_t = 100, N_\pi = 100$) | 0.9750 | 0.6756 | 10.87 |
| Recursive ($N_x = 10, N_t = 50, N_\pi = 50$) | 0.9674 | 0.6572 | 24.47 |
| Recursive ($N_x = 10, N_t = 100, N_\pi = 100$) | 0.9900 | 0.6734 | 97.96 |

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

group is active. The polynomial method in this example can be unstable, with convergence sensitive to starting values and the way that monotonicity, bounds and boundary value conditions are enforced along the path to a solution. Example 5 is from Dharanan and Ellis (2024).[21] 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. Computationally, they report that their approach requires roughly 50 hours of computing time to calculate the solution for this example. Our method obtains a solution, which resembles their approximated solution in slightly over 1 second.

The computation time for our approach depends on $N_x$ and $N_t$. Sometimes a larger value of $N_x$ can result in a reduction in computation, time see Table 1 examples with ($N_x = 30, N_t = 200$). Under Algorithm 2 this would not occur, and each single-agent problem takes longer to compute in these scenarios. However, the iteration towards an equilibrium takes less iterations as the more flexible strategy results in a less coarse approximation of the opposing bid distribution (and associated win probabilities). With the smaller $N_x$, each segment that you move has less of an impact on the win probabilities, leading to a more stable set of adjustments at each step.

Notice that although we construct our solution by moving forward along the bid functions, we will not suffer from the instabilities documented by Marshall et al. (1994). By computing the full value function, rather than proceeding along a single path, as in the solution of the differential equation, we are able to avoid these cases where the wrong path is selected early on, and once selected must be followed. In this way, our global optimization routine avoids compounding early errors along the path. This problem of compounding errors along a path is also an issue in the backward shooting approaches Fibich and Gavish (2011).

The iterated approach requires a choice of tolerance for the stopping criteria. This choice must balance convergence with computation time from performing extra iterations which result in only small improvements. Given the discreteness, of the approximation there is a risk of cycling behavior that can take several iterations to dampen, while not meaningfully impacting the magnitudes of the results when the tolerance is set too tight. On the other hand, when the tolerance is too loose, the predictions can be far from the equilibrium. For the auction application, we set the tolerance so that win probabilities at each price move by less than 1 percent at the final step.

The method can also be applied to common values, interdependent values, and to handle affiliation in the values. It can also be used with other preferences such as risk aversion, and to solve alternative auction formats such as an all-pay auctions, which are difficult to solve with other numerical methods. This extension does not introduce any additional complications and both the iterated and naive approaches can again be applied. The algorithm appears to continue to perform well in those settings.

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.[22] In particular, the computational time

[21] It corresponds to their Example 3.

[22] Indeed, all 5 examples in this section were computed with exactly the same code, which takes as

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. In addition, the step-function approach seems more stable than the polynomial approach as the degree of the approximation $M$ gets large, and as the number of bidder groups increases.

**Full equilibrium algorithm results.** In Table 2, we introduce an additional grid with $N_\pi$ points, representing the probability that the largest opposing bid is below the current price $t$. The use of this grid facilitates the quick solution of the equilibrium fixed point problem. The results highlight several interesting patterns.

First, on computation times. Increasing the fineness of either the $N_t$ or $N_\pi$ grids appear to have similar computational costs. On the other hand, increases in $N_x$ result in larger increases in computation times. The times appear to 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 (we do not acknowledge 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. Increases in the grid of either $N_t$ or $N_\pi$ alone provide little improvement in accuracy, however when both of these grids are more fine, the accuracy improves. In general, the results display an upward bias relative to the iterated solution in the previous section. This is expected given the coarse approximations to the win probability grid, together with the fact that the bidder chooses the best path along the grid. This will tend to cause the rounding error to be magnified in the expected profit metric. However, the resulting strategy functions remain quite similar.

## 3.2 Nonlinear Pricing

In this section we apply our method to study nonlinear pricing games. Our approach can easily incorporate competition, multi-dimensional types, and type-specific participation constraints, features which are difficult to handle under standard approaches.[23] Nonlinear tariffs are widely used by firms to improve efficiency, and to price discriminate by providing consumers with information rents in exchange for revealing information about their types, for instance in regulated markets such as railways and public utilities, as well as airlines, banks, and consumer goods (often via bulk discounts). Similar analysis has been applied to study topics such as optimal taxation and public good provision.

We focus on an application which incorporates competition in schedules. The computational challenges involved with solving the optimal nonlinear pricing schedule have so far limited the empirical work that analyzes this problem. The problem for a monopolist has

input only the number of bidders in each bidder group, the distribution of valuations, and the grid sizes.

[23]For an overview of the parametric utility approach applied to competitive problems see Rochet and Stole (2003), and for the demand profile approach see Chapter 12 of Wilson (1997)

been studied by Luo, Perrigne, and Vuong (2018) who use general preferences but apply an increasing hazard rate assumption to assure that the first-order condition is sufficient. In a discrete setting, this is equivalent to assuming that only the downward constraints are binding. Miravete (2002) and Aryal and Gabrielli (2020) use quasi-linear preferences and since these result in linear demand functions, they provide tractable representations of the solution.

Our approach, on the other hand, requires few assumptions on the distribution of preferences and utility functions, and incorporates all local constraints, both downward and upward. This is possible because we directly calculate the optimal schedule by leveraging the recursive structure of both the constraints and the firm's profit function. As in the auction example, our approach solves for the global optimal schedule rather than the local solution from parametric approximations or differential equation based solution methods.

**Model.** Suppose we are in the nonlinear-pricing model of Example 3 of Section 2. There are two firms, $i = 1, 2$. Consumers have utility function $u(x_1, x_2, t)$, where $x_i = (p_i, q_i)$ are the price and quantity of the bundle of goods they acquire from firm $i$, whereas $t \sim G$ is their type. We again formulate the problem in discrete space with local constraints as in (7). Firm $i$ offers contracts $\hat{x}_{i,n} = (\hat{q}_{i,n}, \hat{p}_{i,n}) \in \hat{X}$ to types $t_n \in T$, with the two-dimensional contract grid being $\hat{X} = ((p_1, q_1), ..., (p_k, q_k), ..., (p_K, q_K))$. The firm has profit function $\Pi_i(\hat{x}_{i,n}, t_n) = \hat{q}_{i,n}(\hat{p}_{i,n} - c)$ and $m_n = G(t_n)$. The set of local constraints, upward and downward, at step $n$, is given by

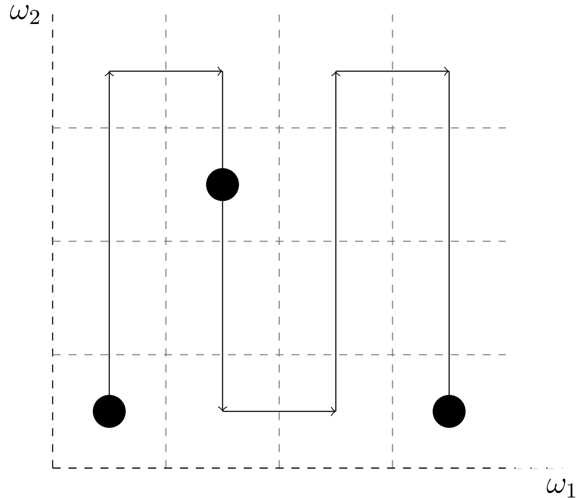
$$\mathbf{R}_i(\hat{x}_{i,n+1}, t_{n+1}; \hat{x}_{i,n}, t_n) = \begin{cases} u(\hat{x}_{i,n}, \hat{x}_{-i,n}, t_n) - u(\hat{x}_{i,n+1}, \hat{x}_{-i,n}, t_n) \geq 0; \\ u(\hat{x}_{i,n+1}, \hat{x}_{-i,n+1}, t_{n+1}) - u(\hat{x}_{i,n}, \hat{x}_{-i,n+1}, t_{n+1}) \geq 0. \end{cases}$$

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

In the multidimensional case, where there is no strict ordering over incentive constraints, the full solution would require a large state-space. To illustrate imagine a 2-dimensional problem. One possible ordering is to choose one dimension, call it $(\omega_1)$, to move along. But then the state must include the contracts offered to all types $\omega_2$ at the previous $\omega_1$ grid point. This is a large state space and with even a moderate sized grid in $\omega_2$ is not computationally tractable. In this case, we solve the problem dropping some of the incentive constraints. We set a grid over types (now a set of rectangular regions), and iterate by snaking up over and down. In each case we store the last state, as well as the last state in our *constraint set*, as illustrated in Figure 3. To begin, the constraint set is empty. We then compute the solution, to the unconstrained problem and check violations of the incentive constraints. For each column, we add to the set of constraints the cell which has the largest violation of the incentive constraints (if any in that column are violated). Then, we repeat. In many cases the solution has few violations with only a small set of constraint states included and this provides a reasonable approximation at a low computational cost.

Note that an alternative way to solve this nonlinear pricing policy would be following the demand profile approach. In that case we would set a price grid and solve for the optimal policy along the quantity grid. This requires stronger assumptions on the demand (to avoid consumers jumping between steps), but is accurately able to find the solution in Example 1 in the subsequent section.

Figure 3: Snaking Picture




Note: This figure illustrates the snaking pattern used to order the two-dimensional partition. The dots indicate the last constraint state that is also held as part of the state space until the next dot arrives.

**Simulations.** This section presents results from two examples. In each, consumers have quadratic utility, with uniformly distributed unknown preference shifts $t \sim U(0, 10)$ $u(q_1, q_2, t) = \frac{\alpha+t}{\beta-\gamma}(q_1+q_2) - \frac{\beta}{2(\beta^2-\gamma^2)}(q_1^2+q_2^2) + \frac{\beta^2}{\beta^2-\gamma^2}q_1q_2$. The results presented set $\gamma = -\frac{1}{3}$, so the products are complements. In addition, $\alpha = 1$, $\beta = 1 + \gamma$. Firm's 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 playing the optimal behavior using the analytic solution. We calculate the utility offered from that firm's menu for every consumer type, and apply our method to solve the single-agent optimization problem for the remaining firm, incorporating this utility as a type-specific participation constraint.
- *Example 2.* Solve for equilibrium in the competitive model where both firms play optimally.

Table 3 presents results for these examples using our approximation, two polynomial approximations, and the true analytic solution.[24] In all three examples the profits are well approximated by our solution method and the step-function approximation of the curves are similar. The polynomial approach produces both higher approximation errors and is much slower in this example. The gap in computational time required between the polynomial approach and our method is much larger in this example than the auction. Given the parameterization in the auction problem, calculating the sum of errors in the differential equation is essentially equivalent to performing two matrix multiplications and taking a difference. In the nonlinear pricing problem however, calculating the profits from a particular price schedule requires figuring out what consumers would choose given that set of offers. This additional complexity adds nonlinearities to the optimization problem

[24]The polynomial approach is discussed in Section 3.1.

Table 3: Nonlinear Pricing – Monte Carlo

| | Profit | | Compute |
|---|---|---|---|
| | Bidder 1 | Bidder 2 | Time (s) |
| *Example 1* | | | |
| Analytic | 5.71 | – | – |
| Polynomial ($M = 1$) | 5.65 | – | 2.20-9.82 |
| Polynomial ($M = 12$) | 5.38 | – | 2.71 |
| Recursive ($N_x = 10$, $N_t = 100$) | 5.64 | – | 0.09 |
| Recursive ($N_x = 10$, $N_t = 200$) | 5.61 | – | 0.19 |
| Recursive ($N_x = 20$, $N_t = 200$) | 5.64 | – | 0.88 |
| *Example 2* | | | |
| Analytic | 5.71 | 5.71 | – |
| Polynomial ($M = 1$, $SV = true$) | 5.88 | 5.62 | 5.63 |
| Polynomial ($M = 1$, $SV = far$) | 5.87 | 4.99 | 14.10 |
| Polynomial ($M = 12$, $SV = true$) | 4.72 | 4.82 | 61.03 |
| Recursive ($N_x = 10$, $N_t = 100$) | 5.64 | 5.71 | 3.58 |
| Recursive ($N_x = 10$, $N_t = 200$) | 5.61 | 5.68 | 6.87 |
| Recursive ($N_x = 20$, $N_t = 200$) | 5.64 | 5.68 | 44.85 |

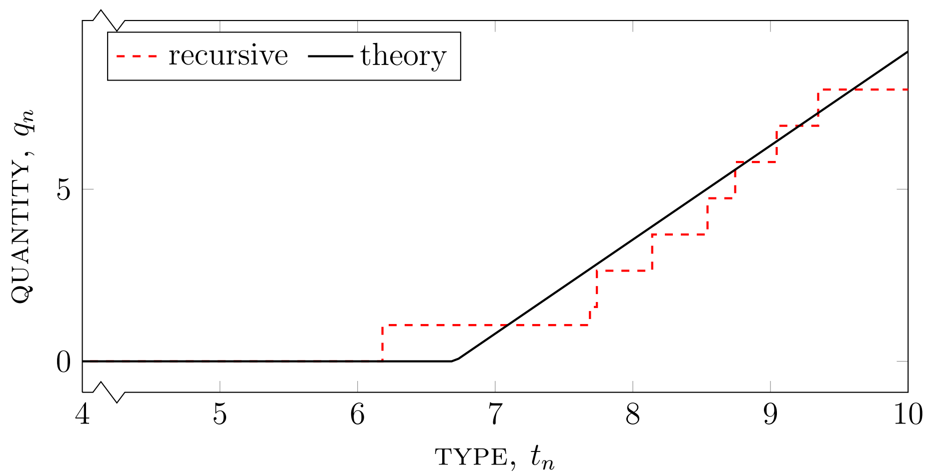
Note: In the competitive case we explore the role of starting values: $SV$ changes the starting values given to the polynomial approach, illustrating the instability of this solution method. Here, *true* indicates the starting values are the true solution, while *far* starts the solution from a [1,-0.01] 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 Iterated Equilibrium Algorithm. Notice that $N_x$ refers to the grid size of both price and quantity, such that the combined grid has a size of $N_x^2$.

for the polynomial approximation and increases the cost of evaluating the errors associated with a given vector of parameters. Figure 4 illustrates the step-function approximation for the equilibrium calculated in Example 2.

**Multidimensional nonlinear pricing.** Multi-dimensional settings are much more complicated to analyze. In the parametric utility approach, integration by parts cannot be used and the problem to solve is a second order calculus of variation problem with an inequality constraint (a so-called *obstacle problem*). The typical approach to working around this problem has been to solve a relaxed problem, then check the remaining constraints and potentially adjust (Rochet and Choné, 1998; Deneckere and Severinov, 2017). Another approach is taken by Ghili and Yoon (2023) who approximate the price schedule as a set of linear functions $p_k q_k$ for $k = 1, 3, 5$ segments. They maximize a non-smooth, objective function with no guarantees on concavity or linearity, using a grid-bisection approach for which convergence to the global maximum is not guaranteed. Rochet and Stole (2003) note that even in a multi-dimensional setting, you can check these constraints by proceeding through a tree of the successive binding local constraints described by a dynamic program. However, in general the set of paths depends on the utility offered at earlier points and so this fact alone does not directly allow for a tractable representation of the optimal price schedule decision problem for the firm.

We explore a problem with multidimensional agent-types, but where the firm has a one-dimensional tool for discrimination–the price offered for $q$ units. Unlike approaches

Figure 4: Competitive Nonlinear Pricing – One-dimensional Types




based on parametrizing and then directly approximating the $p(q)$ function, our approach extends easily to richer contracting environments. We focus on this simple case with a known solution, to evaluate the accuracy of our general approach.

To evaluate the performance of our approach in an environment with multidimensional types we consider an artificial multi-dimensional experiment. In general multi-dimensional environments the true solution is hard to express and so it would be difficult to evaluate the performance of our approach. Focusing on an artificial environment allows us to analyze the performance of the computational method when evaluating a relaxed problem in two-dimensions. Namely, we focus on the problem of a single seller in Martimort and Stole (2009), fixing opponents' behavior at their best responses. In this environment, we assume that bidders draw two-dimensional types, from a common distribution so that the sum of these two variables is the true payoff relevant type, which is uniformly distributed (i.e., $(\omega_1, \omega_2)$ with $\omega_1 + \omega_2 = t$). Crucially we do not allow the algorithm to make use of the fact that there is only one type that matters, but instead must solve the two type version. To solve the optimal price schedule fully in this setting is impossible, for example if the iterative approach moved along the x-axis, the state space would depend on all the vertical grid points at the past point and would therefore be far too large to be tractable. Instead, we consider a relaxation of the seller's problem and iteratively add constraints as discussed above. We stop after the second iteration the number of constraints included becomes very large. This approximate solution results in similar profits: with profits of 6.41 in the relaxed problem, as opposed to the true profits of 5.71. Although we focused on a single-agent problem for illustration, this approach would be easy to iterate in a similar manner to the example with competition in one dimension.

# 4 Application: Department of Transportation Procurement Auctions

A large literature on asymmetric auctions has applied these models to evaluate procurement policies. Many applications use the alternative numerical approaches discussed in

Section 3, which, as well as being numerically unstable, are computationally slow. This computational constraint limits the set of counterfactuals that can be analyzed, and could be important for cases such as location choices, entry, 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 approach.

## 4.1 California Bid Preference Program

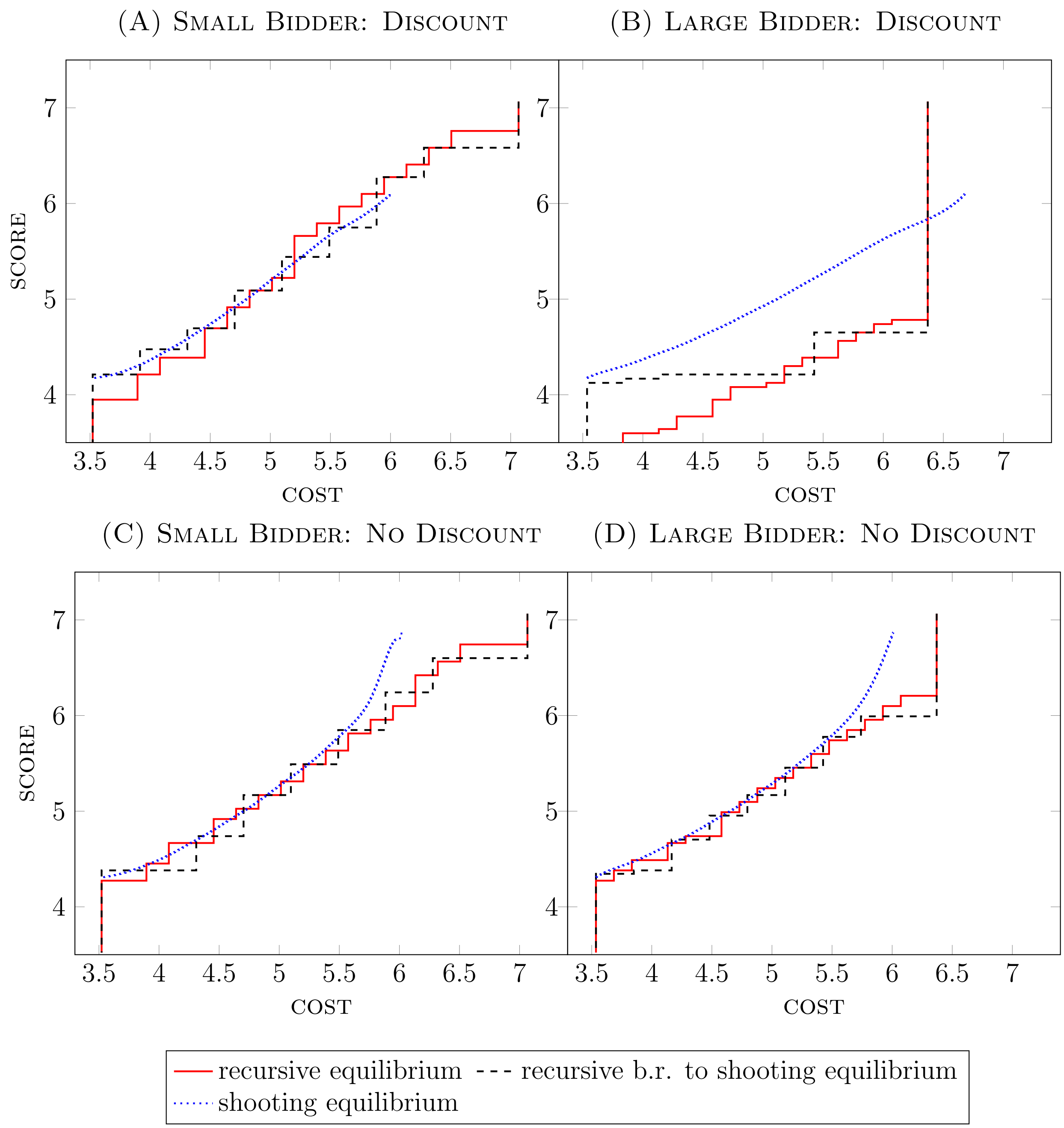
Bid preference programs are popular policies which are in wide-spread use across the US. These policies offer a subset of bidders the opportunity to pay only a fraction of their bid, and therefore introduce asymmetry and potentially uncommon supports, even in settings where bidders are initially similar. As a consequence, boundary conditions and challenges with shooting algorithms could be important in this setting, especially when evaluating counterfactual preference policies that include large discounts or in settings with several heterogeneous bidders. In the case of two bidders the relevant boundary conditions are easy to derive. However, with multiple bidders there is a potential for bid-bifurcation, as highlighted by Hubbard and Kirkegaard (2015). So far, this has not been accounted for in the literature. For example, the left boundary condition for two bidders is used in Rosa (2019), Hubbard and Paarsch (2009), Krasnokutskaya and Seim (2011) and Marion (2007). By imposing this left boundary condition, these papers rule out the possibility of bid bifurcation, and may understate competition especially when there are several strong bidders.

To investigate the implications of this restriction, we use the setting of Krasnokutskaya and Seim (2011) who study of the California Small Business Preference program in the context of California Department of Transport procurement auctions for highway and street maintenance. We focus on the first auction in their sample, with project ID 1177. We use our method to highlight cases in their empirical examples where bid bifurcation could arise and could result in meaningful changes in the profit predictions. Our method also provides a substantial reduction in computation time, cutting the average time to compute an equilibrium by over 90%. In addition, we provide examples where their shooting algorithm does not fully converge but stops with terminal values to the left of the right boundary.[25] Both bid bifurcation and failure to fully converge can have a meaningful impact on bidder profits.

Figure 5 compares their shooting algorithm to our method in two examples. In both examples, large bidders have costs on $[3.54, 6.37]$ whereas small bidders have costs on $[3.52, 7.07]$. Panel A and Panel B of Figure 5 presents an example auction with 3 large bidders who have an initial cost advantage, and on top of this receive a subsidy of 30% and two small bidders. The solid lines indicate the equilibrium strategy computed using our recursive algorithm, the dotted lines the equilibrium computed using the shooting algorithm from Krasnokutskaya and Seim (2011), and the dashed lines the bidders' best response as calculated by the recursive algorithm when the opponents behave according to the equilibrium strategy computed using the shooting algorithm. We chose this example as bid bifurcation is likely to occur in this scenario, which would drive a difference between

[25]This behavior occurs also in some cases of Somaini (2020) but in that setting the problem appears to be much less pronounced.

Figure 5: Bid Strategies – Preference Program



our results and their solution. Note that our solution contains bid bifurcation: the three competitive large bidders continue bidding at lower levels and compete each others' bids down, even after all small bidders have dropped out. This bifurcation is not present in the shooting algorithm solution, since it imposes a boundary condition which rules this out.[26] Panels C and D present an example using the same auction with no discount. In this case, the two groups of bidders are sufficiently similar to one another, and the equilibrium does not involve bid bifurcation. However, the shooting equilibrium strategy appears to end early and fails to satisfy the right boundary condition. This behavior looks similar to what one might expect given the failures of shooting described in Fibich and Gavish (2011).

To illustrate that this could matter for conclusions, Figure 6 shows bidder profits computed using our solution method and their method for the same example auction highlighted above under a policy intervention consisting of a negative discount. In the example large bidders are initially strong (low cost), and a negative discount corresponds to a subsidy to large bidders. There are 2 small and 3 large bidders. The dotted and loosely dotted lines represent the solution computed using the shooting based solution method of Krasnokutskaya and Seim (2011), while the solid and dashed lines represent our solution. We also include dash-dotted and dash-dot-dotted lines that show the profit from the recursive algorithm best response to the opponents playing the shooting algorithm equilibrium strategy. This indicates that at most discount levels the bidder can profitably deviate from the shooting strategy. The recursive best response profits are slightly lower in a few cases, which we believe is due to our strategy lying in a much more restricted space, and being subject to numerical error in integration on the coarse grid. However, away from these cases, our best response strategy shows large profit improvements.

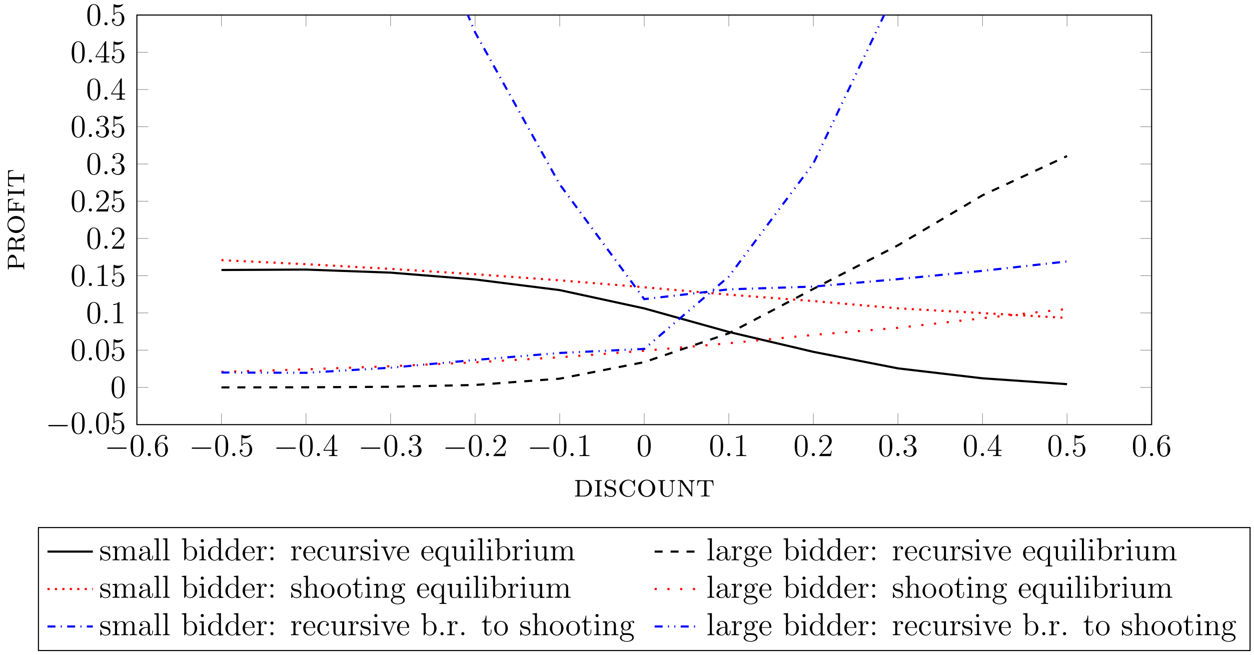
## 4.2 Michigan Highway Procurement Auctions

We next apply our method to the Michigan highway procurement auctions analyzed by Somaini (2020), which feature interdependent values. This allows us to evaluate the performance of our solution outside the private values framework. The Michigan Department of Transport uses first-price sealed-bid auctions to award construction and maintenance contracts which pay a fixed amount to the winning firm, regardless of the realized cost of executing the contract. Somaini (2020) models cost as a linear function of observable project characteristics, bidder distance to project, and the (privately observed) signals of all bidders. Thus, costs are interdependent and bidding will in general suffer from the winner's curse.

We also consider perturbations of his data generating process, that move us away from the linear, normal case. In particular we consider perturbations to costs that increase curvature, $c_i = \delta c_p + (1-\delta) l_i (c_w s_i)^{c_e}$, where $s_i$ is the signal in quantile space, $c_p$ the costs from his model, $\delta$, a weight between his cost and the additional component, $l_i$ and $c_w$

[26] The bids are shown in stated dollars, so that actual payments for large bidders in this panel are 1.3 times the amount in the figure. The right boundary condition for strong bidders should therefore end at 4.9. Krasnokutskaya and Seim (2011)'s left boundary condition forces the quoted bids (rather than paid) to match at the left end point which precludes bid bifurcation. Notice that their solution for the large bidder exceeds the maximum cost of 6.37. We believe this is an artifact of the well-known problems with the shooting algorithm and the fact that they solve in inverse strategy space.

Figure 6: Bidder Profits – Subsidy Variation




are linear coefficients on the bidders' own signal, and $c_e$ an exponential term to induce curvature and move the perturbed version away from the linear, normal set-up. This version is convenient as it nests examples 1, 3 and 4 from our Monte Carlo exercise, (e.g. for example 1, set $\delta = 0$ and $l_i = c_w = c_e = 1$), allowing us to evaluate the performance on a problem with a known solution, before exploring the distributions that are relevant in the data. It also nests his model (set $\delta = 1$). We anticipate that by adding nonlinear terms to the cost function, convergence will be more difficult to achieve than in his baseline setup.

We first measure baseline performance by using his algorithm directly to compute a symmetric auction with uniform private values on [0,1] with different numbers of bidders. Table 4 compares the speed and accuracy of the two solution methods. Across different numbers of bidders, our method appears to provide a more accurate approximation of the profits earned by bidders. In addition there is a substantial difference in computation times, with the shooting based algorithm both starting out slower and slowing down at a faster rate relative to our method as $N$ increases. The computation times depend on the distribution of costs in the auction of interest, and 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 poses a limitation for the counterfactuals that can be considered.[27]

[27] Somaini (2020) overcomes this by considering a sub-sample of 250 randomly selected auctions with between 2 and 10 bidders, and using a grid of 50 prices for reserve prices. Because he assumes that only participants with probability of participating of at least 0.07 will participate, he rarely needs to solve the model with a large numbers of bidders. This saves considerable computation, in an area of the parameter space that would otherwise be slow to compute.

Table 4: Somaini: Uniform Private Values – Algorithm Comparison

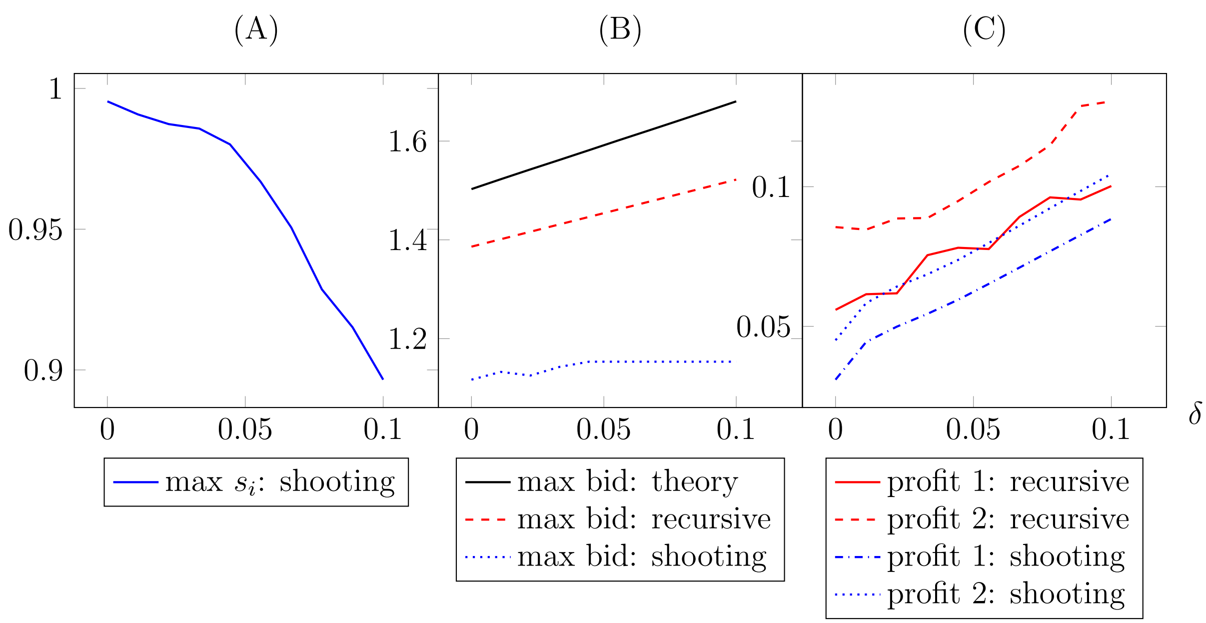
| | Analytic | Shooting | Recursive |
|---|---|---|---|
| N-bidders=2 | | | |
| Time (s) | - | 2.45 | 0.54 |
| Expected Profits | 0.167 | 0.153 | 0.167 |
| N-bidders=3 | | | |
| Time (s) | - | 7.45 | 0.82 |
| Expected Profits | 0.08 | 0.073 | 0.079 |
| N-bidders=4 | | | |
| Time (s) | - | 13.85 | 1.39 |
| Expected Profits | 0.05 | 0.042 | 0.045 |
| N-bidders=5 | | | |
| Time (s) | - | 28.80 | 1.63 |
| Expected Profits | 0.03 | 0.026 | 0.029 |
| N-bidders=6 | | | |
| Time (s) | - | 46.81 | 2.39 |
| Expected Profits | 0.02 | 0.022 | 0.021 |

Note: *Shooting*: The shooting algorithm of Somaini (2020). *Recursive*: the Recursive Iterated Equilibrium Algorithm.

We then explore the sensitivity of his approach to perturbations of the form described above. In particular, we take a random auction from the data, and compute the equilibrium for different values of delta. To make it more challenging for the algorithm, throughout we 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 fix the number of bidders at 2 for this exercise.

For the shooting algorithm to converge, the maximum signals should be close to 1, the upper bound of the support of the uniform distribution. With only two bidders, the right boundary condition should solve the optimization problem for the weak bidders' choice of a bid above their cost by trading off expected surplus against the win probability, given the weak types' signal distribution. Figure 7 presents results on this. Panel A illustrates the maximum signal reached in the shooting algorithm as a function of $\delta$. The maximum signal reached should always be 1, but we observe a steady downward trend and at $\delta = 0.1$ the maximum signal reached is below 0.9. Panel B presents the maximum reached as well as the maximum cost of the advantaged bidder types, as a function of $\delta$. The maximum bid from the shooting algorithm undershoots the cost substantially. The maximum bid from our method slightly undershoots the cost as well, due to the density of the grid. Indeed, the difference shrinks as the number of grid points is expanded. On the other hand, with a fixed number of grid points, the undershooting gap widens with $\delta$. This is due to the fact that with a fixed number of points, the grid becomes more coarse when $\delta$ increases. In panel C we include the approximated profits, which highlight that these differences are empirically relevant, with the profits predicted from the shooting algorithm on average only 75% of those from our approach. In addition, in with our method, unlike with a differential equation approach, errors at the starting point will not mean that a

FIGURE 7: EVALUATING CONVERGENCE – PERTURBED DGP




wrong path is followed throughout the entire solution.

# 5 Application: Cereal Product Sizes

Finally, we apply our method to study the choice of the menu of box sizes offered for different cereal products. For this exercise, we use demand estimates obtained using the retail scanner and consumer panel datasets provided by NielsenIQ. This menu design relates to recent policy concern over observed reductions in some box sizes, which has been labeled 'shrinkflation', and negative press coverage of cereal companies selling smaller boxes.[28] While these concerns could be important if they reflected increases in market power or if consumers were unaware of the changes,[29] the adjustments in the menu could also result in increases in consumer welfare, and banning the practice may make consumers worse off. Over the last 20 years cereal companies have regularly adjusted the packages sizes that they offer, tailoring prices and package sizes to market conditions. In fact, similar concerns about shrinking box sizes were raised in 2007.[30]

In many industries firms offer multiple products, and the characteristics of their product line change over time. Empirical studies of this have documented how characteristic

[28]For example: McNair (2023), or `https://www.weforum.org/agenda/2022/03/how-companies-are-hiding-inflation-without-charging-you-more/`. There was even a bill introduced in congress to try to prevent this practice: `https://www.casey.senate.gov/imo/media/doc/shrinkflation_report.pdf`.

[29]In surveys (and in media discussion) consumers do seem to be aware of the change in package sizes (`https://www.theguardian.com/news/ng-interactive/2024/mar/26/how-shrinkflation-is-impacting-your-cereal-and-making-you-pay-more-for-less`).

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

choices and the number of options offered relate to competition (McManus, 2007; Berry and Waldfogel, 2001; Sweeting, 2010; Seim and Viard, 2011; Borzekowski, Thomadsen, and Taragin, 2009; Watson, 2009) and have identified those characteristics most important for price discrimination (Draganska and Jain, 2005, 2006). In addition, models of product entry have been used to study whether to offer discrete options (Draganska, Mazzeo, and Seim, 2009; Fan and Yang, 2020; Wollmann, 2018) and what characteristics of a single product to offer (Fan, 2013; Crawford and Shum, 2007; Crawford, Shcherbakov, and Shum, 2019).

However solving for the set of varieties that a firm chooses to offer in counterfactuals has been a challenging problem. This is due to the size of the set of potential products that can be offered typically being very large. While the set of possible characteristics for each individual product (e.g. sizes of a cereal box) is large, the number of possible menus is exponential in the number of products that the firm could offer. Because of this, the existing literature restricts the set of feasible choices to a small number and solves the simplified game. For example, Fan and Yang (2020) model the set of potential cell phones as those sold by the firm plus two additional products to fill gaps in the quality spectrum, Wollmann (2018) reduces trucks to a small number of combinations of cab style and weight rating, Eizenberg (2014) studies four product types for computers, and Draganska et al. (2009) model a product entry game for small set of extra optional ice cream flavors. While this approximation might be reasonable when sizes or characteristics are limited by other constraints, this approach is not likely to perform well in many settings, including cereal markets, where a great deal of variation in the set of offered weights is observed, even at the level of fractions of an ounce. In addition, the firms offer many menu options. This means that using alternative non-tailored optimization tools at each step of the best response iteration (e.g. as used to solve two menu points in Fan (2013)) would require nesting a complex optimization problem. Our method simplifies this optimization considerably and easily allows for the number of points to be endogenous.

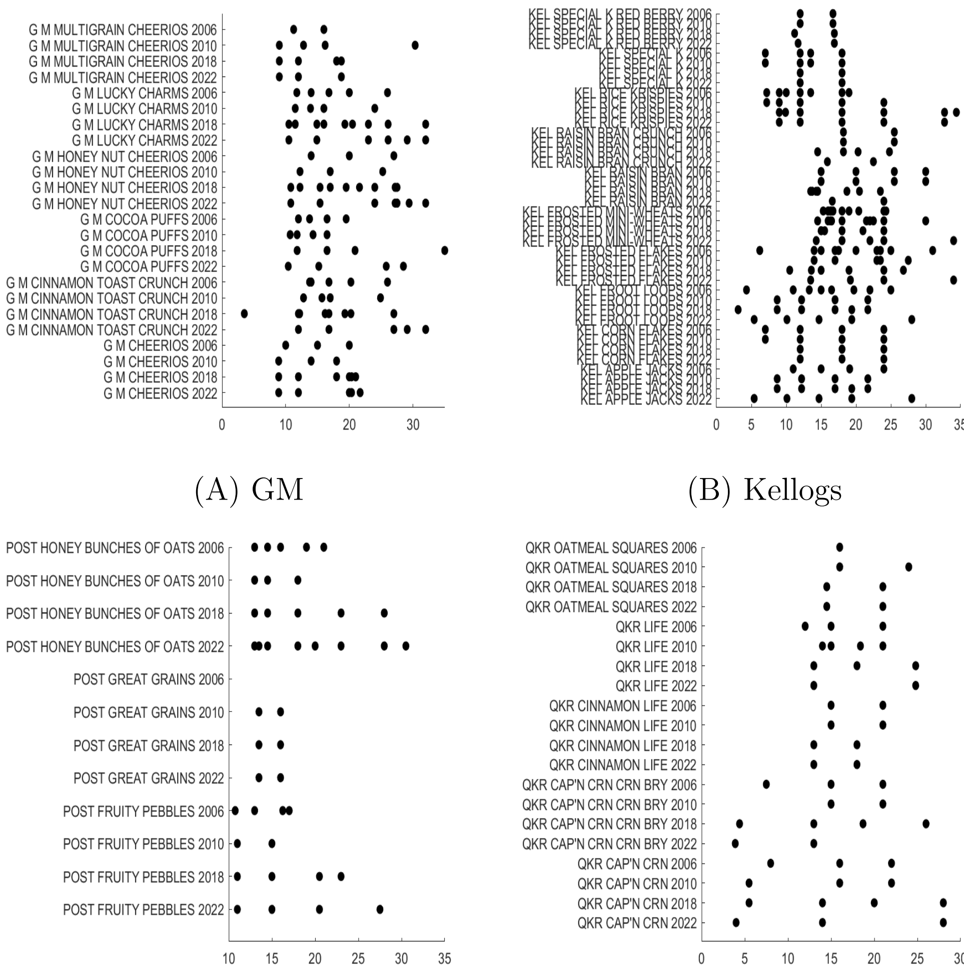
Our method allows us to capture both horizontal and vertical differentiation, and we can apply the method to solve for counterfactuals given similar demand models to those used in many existing papers including Fan (2013), McManus (2007) and Fan and Yang (2020). We require two assumptions on the demand model in order to apply our method to compute counterfactuals. First, that package specific shocks occur at the product level not the product-size level, thus by adding an additional package the firm does not get an additional draw of the idiosyncratic consumer component of utility. Second, for computational tractability, we require that the menu adjustments take place across one dimension (e.g. package size) and that preferences for this item across consumers can be described by a one dimensional type (e.g. taste for size). In the presence of multiple dimensions of preferences over the sizes, or if there were a second trait that could also be varied by a firm within a product, we could extend the methods used in the multidimensional extension in Section 3.2, however these approaches are less accurate and more computationally costly.

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 3.2. However, in a setting with fewer product lines to be simultaneously calculated, the full solution approach could also be applied.

## 5.1 Data and Background

The set of cereal boxes offered on the market has long been changing. Figure 8 plots the menu for the 25 most popular brands with each subplot depicting the set of products sold by each firm for each year. Two things especially are worth noting. First, the box size seems to be a flexible choice, with many different unique box sizes used over time (in contrast to products such as soft drinks where can size has remained quite constant). Second, it is difficult to detect a persistent pattern towards shrinking boxes as discussed in the media.

FIGURE 8: REAL BOX SIZES




(A) GM

(B) Kellogs

(C) Quaker

(D) Post

Note: Figure includes all sizes that have at least 1% of the products sales in that year.

In order to provide a simple summary description of the changes between 2018 and 2022, we divide the menu for each product into three groups, small boxes (below 12 oz), medium (12-18 oz) and family boxes (18-24 oz).[31] In 2022, small boxes are offered by 50

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

% 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 a medium sized format, and the average size of a median box shrunk from 14.97 oz to 14.72 oz. In the family 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.

## 5.2 Demand Estimation

We specify the following model of demand:

$$u_{ijkt} = \beta p_{jkt} + \beta_i size_{jk} + \xi_{jkt} + \omega_j + \epsilon_{ijkt}, \tag{10}$$

where $\omega$ is a brand-level fixed effect. The brand-level fixed effects will absorb all the characteristics unless we want to include random coefficients. The specification has a random coefficient on size only, which is also a characteristic at product level. With consumer demographics data, we can further specify:

$$\beta_i \sim \mathcal{N}(\Pi d_i, \Sigma\Sigma'), \tag{11}$$

where $d_i$ is demographics (including a constant) and $\Pi$ and $\Sigma$ are parameters that we will estimate. We make two additional assumptions on the form of preferences and costs across package sizes. In the counterfactuals we need to know costs $c(q)$ and preferences $\xi_{jkt}$ at $q$ not submitted in the data. To allow for this, we model the preferences by linearly interpolating between submitted $q_k$, and as flat when extending the range, and we model marginal costs as a quadratic function $mc(q) = a_1 + a_2 q + a_3 q^2$.

Estimation follows Backus, Conlon, and Sinkinson (2021) and Nevo (2000b), for details see Appendix D. 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 fit the model of costs across package sizes, we fit the quadratic function discussed above to the unrestricted estimates of marginal costs. In order to improve the estimation performance for costs, and to handle firms that do not offer any large packages, we enforce a restriction when solving for the parameters $a$. In particular, we restrict the marginal cost at the largest $q$ on the $q$-grid used for counterfactuals to be large enough so that the biggest package that the firm wants to offer is the one that they are observed to offer in the data. This helps to reduce the impact of noise in the finite sample estimates of marginal costs when marginal costs are only observed at small $q$ relative to upper bound of the support of $q$.

Finally, we estimate bounds on fixed costs, by using our method to compute the menu offered by the firm against the current menus of the opponents, and then finding the package submission cost such that the number of packages they choose to offer in the computed equilibrium matches the number offered in the data.

## 5.3 Evaluating Model Fit

We first solve for the menu for the game played in the data to provide a benchmark that allows us to assess the impact of our parametric restrictions. While solving counterfactuals we solve for equilibrium in a single randomly selected market in September of 2018. In all counterfactuals we assume that there is a single $\epsilon_{ijt}$ rather than $\epsilon_{ijkt}$. This means that firms do not want to introduce additional package sizes to only gain an additional draw of the idiosyncratic shock.

The baseline computation for the full menu sold of each product does a reasonable job approximating the menus. The total profits and their distribution across firms are similar to the profits implied by the set of packages offered in the data, however the estimates overshoot slightly on the profits from small packages. As in the previous section, summarizing the results by the changes in small, medium and large packages, we find a predicted share of 64% of products offer small packages relative to 60% in the data. These have an average size of 9.94, relative to 10.55, in the data. 88% of firms offer medium packages, relative to 76% in the data with an average package size of 15.20 in data and 15.96 in the computed solution. Finally, in the computed solution, 76% of products offer a large option, compared to 52% in the data, with the large package on average at 21.42 in the computed solution and 20.93 in data. The computed menus also use too many steps with 2.96 steps on average, as opposed to 2.12 in the data.

For most products GM offers a menu with three options, the notable exception being Reese's puffs. Consumers have similar levels of taste across GM's products. This differs from Kelloggs which has a higher variance both in the number of products offered and in their sizes. Quaker offers only 2 product sizes but there is some variation in the size of their small and large boxes. The model does a good job predicting this cross-product variation.

It is possible that the iterated approach to jointly optimizing the menus across products within a firm could find local optima. For example, if one product is much more profitable only when sold alone, it may be difficult to discover this solution when it is priced after fixing an offering of other products. To account for this we recompute the equilibrium using a variety of product orderings for firms as they optimize their menus. Of particular interest, we use orderings based on consumers tastes for the product, and in terms of costs, and in both cases we arrive at the same solution as the random initial ordering.

To explore the stability of the iterated approach, we can analyze the sensitivity of the set of packages offered to changes in the expected level of utility offered to consumers of each type from other goods. This captures how much a firm would respond to a small change in menu by one opponent. In particular, we fix a product (here GM Cheerios) and draw 100 curves uniformly at random for the utility offered by opposing firms that are within a distance of: 0.05, 0.25 or 1 of the type-specific maximum utility actually offered by the opponents and for each of these compute the chosen package sizes and compare these to those offered in the data. The corresponding sum of squared differences to between predicted and chosen price and quantity pairs are all zero, suggesting that the equilibrium is reasonably stable.

## 5.4 Counterfactuals

The first counterfactual exercise that we consider loosely mimics the calculations that may have been involved with changes in box-sizes during the discussion of shrinkflation. We evaluate a 10% increase in production costs, for all firms, and track the resulting changes in sizes and prices of the boxes offered. To isolate the impact of the adjustment of package sizes, we compare this against a setting with fixed box sizes. The second counterfactual exercise follows Nevo (2000a) and considers a hypothetical merger between Quaker and GM, as well as Quaker and Kelloggs.

**Cost Increases.** The full set of quantities offered after a change in costs are presented in Figure A.1. As costs increase, slightly less products tend to be offered as part of each product-line, falling to an average of 2.56 from 2.84. 56% of firms offer a small box, down from 64% before with the average small box size increasing to 10.11oz. There is also some growth in the size of medium boxes, to 15.5 oz on average from 15.2 oz and 84% of firms offer a medium box from 88% in the baseline. In addition, somewhat less 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.

Allowing for the box size adjustment also has important implications for consumer surplus and profits. When we allow for changes in the box sizes, GM profits fall by 6%, and Kelloggs' profits fall by 5% . On the other hand, Post profits rise by 1% and Quaker fall by 13%. Consumer surplus falls by 5.4%. With fixed box sizes, GM profits fall by 9.6%, Kelloggs' profits fall by 5.1%, Post fall by 2.2% and Quaker fall by 9.0%; consumer surplus falls 7.5%. 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. There are important distribution implications from this omission across the firms and profits, especially for Kelloggs, Post and Quaker are dramatically different in these two models.

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, increasing by more than 25% the loss of consumer surplus. In addition, such a ban could have important competitive impacts across firms, providing some with advantages while harming others. These implications must be considered in the discussion around limiting the impacts of shrinkflation.

**Hypothetical Merger.** In addition, the results suggest that allowing for package adjustment should be important when evaluating other policy counterfactuals, including evaluating the effect of mergers (Nevo, 2000a). Table 5 shows the changes in profits from two hypothetical mergers.

Table 5: Profit Changes from Merger

| Merger | Fixed Quantity Remaining % Δ Profits | | | Adj. Quantity Remaining % Δ Profits | | |
|---|---|---|---|---|---|---|
| | GM | Kelloggs | Post | GM | Kelloggs | Post |
| Quaker and GM | 27.3 | 3.1 | 4.1 | 30.1 | 2.1 | 12.1 |
| Quaker and Kelloggs | 4.6 | 18.5 | 5.7 | 4.8 | 17.2 | 14.12 |

In a merger of GM and Quaker, GM profits rise by more than the direct effect of Quaker's initial profits, with modest gains for Kelloggs and important gains for Post. When box sizes are held fixed, the gains for Post are much smaller. The small and medium box sizes offered by firms are quite comparable. Fewer products (68% down from 76%) offer a large box and, conditional on offering it, the size falls to 20.89 oz from 21.43 oz.

In the simulated merger of Kellogg and Quaker the results are similar. The trends of changes in size are similar to the merger with GM, however there is some growth in the average size of a medium box, up to 15.39 oz from 15.19 oz, and a decrease in average large box size to 20.89 oz from 21.43 oz. The most notable changes in the menu come from the newly merged firm, which expands its medium sized offerings and cuts back on the large box sizes offered for Quaker's former products.

# 6 Conclusion

This paper introduces a method for computing counterfactual equilibria in pricing games. The key observation is that in many problems the function that we are trying to solve for, such as a tariff function in price discrimination games or a bid function in auctions, can be arbitrarily closely approximated by a finite-dimensional step function, and the individual segments in this approximation can be solved for by formulating a sequential program. The sequential formulation allows us to find the *global solution* in the set of step-functions very quickly. This presents a major advantage over other methods for which local optimization is often easy, but global optimization costly.

We provided a set of simulations that highlight the performance of this method for computing equilibrium of Asymmetric Auctions. Because of the form of differential equation in the auction problem, shooting algorithms for solving the differential equations are well known to be unstable in these problems. Our approach instead searches sequentially over the best path to each point, avoiding the instabilities associated with differential equation approaches that follow a single path. Perhaps the most successful approach for this problem to date has been polynomial approximation tools, however these use a local search over parameters, and can become caught at local minima and can propagate numerical errors. By using the set of step functions as our basis functions, we do not need to rely on local minimization but can use fast global optimization tools. We also applied the method to nonlinear pricing problems including a multidimensional nonlinear pricing problem, where we omitted a carefully selected subset of the constraints. The method appears to perform well in this setting as well, outperforming other polynomial approximation tools.

Finally we presented an application to an endogenous product choice game. We estimated a demand model on Ready-to-eat breakfast cereals, and then asked firms to choose both a box size and price. Following the recent discussion of shrinkflation, we considered a counterfactual where firms face a 10% increase in costs and recomputed the equilibrium set of boxes and prices. The model has very different implications for profits, and consumer surplus than a model where firms are only allowed to adjust prices. This has important policy implications, and would suggest that banning box-size adjustment in response to perceived shrinkflation could double the loss in consumer surplus and is therefore a terrible policy proposal. In addition, the profit impacts across firms are very different in the two models, suggest that it might be important to allow for box-size adjustment when evaluating merger proposals.

# Appendix A Proofs

*Proof of Proposition 1.* Fix a discrete pricing game with local constraints described by $(\hat{T}, \hat{Y}, \Pi, m_n, \mathbf{R}, \hat{\mathcal{X}})$. We wish to show that the set of solutions to the maximization problem (2), $\mathbf{X}^\dagger$, is equal to the set of solutions of Algorithm 1, $\mathbf{X}^*$. Recall that at any step $n$, we use $y$ to denote the action at step $n$ and $z$ to denote the action at step $n+1$.

*Step 1: Show equality of constraints.* We first show that the set of constraints are the same. In the maximization problem (2), any potential solution $\mathbf{x}'$ must satisfy constraint (3):

$$\mathbf{R}(x'_n, t_n; x'_{n+1}, t_{n+1}) \geq 0 \text{ for all } n < N.$$

On the other hand, in Algorithm 1, any potential solution $\mathbf{x}''$ must satisfy constraint (8) evaluated at $z = x''_{n+1}$ and $y = x''_n$:

$$\mathbf{R}(x''_n, t_n; x''_{n+1}, t_{n+1}) \geq 0 \text{ for all } n < N.$$

Hence, the set of constraints is the same.

*Step 2: Rewrite maximization problem.* Let $\mathcal{C} \subset \hat{\mathcal{X}}$ be the set of step-functions that satisfy the local constraints. For $\hat{\mathbf{x}} \in \mathcal{C}$ let

$$W(\hat{\mathbf{x}}) \equiv \sum_{n'<N} \Pi(\hat{x}_{n'}, t_{n'}) m_{n'}(\hat{x}_{n'}, \hat{x}_{n'+1}) + \Pi(\hat{x}_N, t_N) m_N(\hat{x}_N).$$

where we have used Definition 1 to replace $\mu$ by $m$.

The discrete pricing game (2) with local constraints can be written as $\max_{\hat{\mathbf{x}} \in \mathcal{C}} W(\hat{\mathbf{x}})$. For $1 \leq n < N$, collect the first $n$ terms to define

$$v_n(\hat{x}_1, ..., \hat{x}_n | \hat{x}_{n+1}) \equiv \sum_{n' \leq n} \Pi(\hat{x}_{n'}, t_{n'}) m_{n'}(\hat{x}_{n'}, \hat{x}_{n'+1}),$$

and then collect the last $N - n$ terms to define

$$w_n(\hat{x}_{n+1}, ..., \hat{x}_N) \equiv \sum_{n<n'<N} \Pi(\hat{x}_{n'}, t_{n'}) m_{n'}(\hat{x}_{n'}, \hat{x}_{n'+1}) + \Pi(\hat{x}_N, t_N) m_N(\hat{x}_N).$$

This allows us to separate $W(\hat{\mathbf{x}})$ into two parts for an arbitrary $1 \leq n < N$:

$$W(\hat{\mathbf{x}}) = v_n(\hat{x}_1, ..., \hat{x}_n | \hat{x}_{n+1}) + w_n(\hat{x}_{n+1}, ..., \hat{x}_N),$$

Suppose $n < N$ and let $\hat{\mathbf{x}}_n$ be the first $n$ elements of $\hat{\mathbf{x}}$ with $\mathcal{C}_n(z)$ the set of $\hat{\mathbf{x}}_n$ corresponding to $\hat{\mathbf{x}} \in \mathcal{C}$, conditional on $z$. Define $\mathbf{X}_n^\dagger(x_{n+1}^\dagger)$ as the set of $\mathbf{x}_n^\dagger$ that are the $n$'th first elements in some vector $\mathbf{x}^\dagger \in \mathbf{X}^\dagger$ with $x_{n+1}^\dagger$ as its $n+1$'th element. Thus,

$$\mathbf{X}_n^\dagger(x_{n+1}^\dagger) = \underset{\hat{\mathbf{x}}_n \in \mathcal{C}_n(x_{n+1}^\dagger)}{\operatorname{argmax}} \; v_n(\hat{\mathbf{x}}_n | x_{n+1}^\dagger). \tag{A.1}$$

*Step 3. Show equality of solution sets conditional on next step.* We now proceed by way of induction. Define $X_n^\dagger$ as the set of $x_n^\dagger$ that are the $n$'th element in some vector

$\mathbf{x}^\dagger \in \mathbf{X}^\dagger$. Let $\bar{X}_n(\tilde{x}_{n+1})$ be the set of $\bar{x}_n(\tilde{x}_{n+1})$. Furthermore, let $\bar{\mathbf{X}}_n(\tilde{x}_{n+1})$ be the set of $\tilde{\mathbf{x}}_n = (\tilde{x}_1, ..., \tilde{x}_n)$ such that $\tilde{x}_k \in \bar{x}_k(\tilde{x}_{k+1})$ for $k < n+1$.

Base case: $n = 1$. We wish to show that $\bar{\mathbf{X}}_1(x_2^\dagger) = \mathbf{X}_1^\dagger(x_2^\dagger)$ for all $x_2^\dagger \in X_2^\dagger$. Observe the following:

$$\begin{aligned}\bar{X}_1(z) &\equiv \underset{y \in C_1(z)}{\operatorname{argmax}} \{\Pi(y, t_1) m_1(y, z)\} \\ &= \underset{\hat{x}_1 \in C_1(z)}{\operatorname{argmax}} \{v_1(\hat{x}_1 | z)\}\end{aligned}$$

Since $\bar{\mathbf{X}}_1(z) = \bar{X}_1(z)$, equation (A.1) then implies the result.

Induction Step: $1 < n < N$. Now, suppose that $\bar{\mathbf{X}}_{n-1}(x_n^\dagger) = \mathbf{X}_{n-1}^\dagger(x_n^\dagger)$ for all $x_n^\dagger \in X_n^\dagger$. Fix $\mathbf{x}^\dagger = (x_1^\dagger, ..., x_N^\dagger) \in \mathbf{X}^\dagger$. Recall the definition of $V_n$:

$$V_n(x_{n+1}^\dagger) \equiv \max_{y \in C_n(x_{n+1}^\dagger)} \left\{ V_{n-1}(y) + \Pi(y, t_n) m_n(y, x_{n+1}^\dagger) \right\}.$$

Observe that if $\tilde{\mathbf{x}}_n \in \bar{\mathbf{X}}_n(z)$, then

$$V_n(z) = v_n(\tilde{\mathbf{x}}_n | z).$$

Let $z = x_{n+1}^\dagger$. By our assumption, $\mathbf{x}_{n-1}^\dagger \in \bar{\mathbf{X}}_{n-1}(x_n^\dagger)$, so we can attain $v_n(\mathbf{x}_n^\dagger | x_{n+1}^\dagger)$ by setting $y = x_n^\dagger$, and by (A.1) this must maximize $V_n(x_{n+1}^\dagger)$. Hence, $V(x_{n+1}^\dagger) = v_n(\mathbf{x}_n^\dagger | x_{n+1}^\dagger)$. It follows that $\mathbf{X}_n^\dagger(x_{n+1}^\dagger) \subset \bar{\mathbf{X}}_n(x_{n+1}^\dagger)$.

Now suppose that there exists $\tilde{\mathbf{x}}_n \in \bar{\mathbf{X}}_n(x_{n+1}^\dagger)$ with $\tilde{\mathbf{x}}_n \notin \mathbf{X}_n^\dagger(x_{n+1}^\dagger)$. By the previous observations, this would imply $v(\tilde{\mathbf{x}}_n | x_{n+1}^\dagger) \geq v(\mathbf{x}_n^\dagger | x_{n+1}^\dagger)$. But then $\tilde{\mathbf{x}}_n \in \mathbf{X}_n^\dagger(x_{n+1}^\dagger)$, a contradiction. It follows that $\bar{\mathbf{X}}_n(x_{n+1}^\dagger) \subset \mathbf{X}_n^\dagger(x_{n+1}^\dagger)$.

Hence, $\bar{\mathbf{X}}_n(x_{n+1}^\dagger) = \mathbf{X}_n^\dagger(x_{n+1}^\dagger)$.

*Step 4. Final step.* Let $n = N$. We can then repeat induction step of step 3, substituting $v_n$ for $W$, $\mathbf{X}_n^\dagger(x_{n+1}^\dagger)$ for $\mathbf{X}^\dagger$ and $\bar{\mathbf{X}}_n(x_{n+1}^\dagger)$ for $\mathbf{X}^*$. This proves the result.

□

# Appendix B Algorithm Implementation

In this section we discuss the implementation of the algorithm. In particular, we discuss how to write down the economic problem so as to choose the space of integration conveniently, how the choice of grid size affects the approximation of the discretization, and finally we discuss other issues such as smoothing parameters.

## B.1 Choice of Space for Integration

There may be multiple ways to write a given economic problem in the form of equation (1), and the choice of variables matters for the implementation and tractability of the algorithm. For example, for a bidder at an auction, $x(t)$ could be a strategy function, mapping private type $t$ into a bid, or it could be an inverse strategy function mapping bid $t$ to an associated threshold type. We highlight the practical considerations when determining whether our tool can be applied to a given problem, and if multiple formulations of the problem are possible, we provide guidance on which should be adopted.

First, our algorithm relies on a representation where only local constraints $\mathbf{R}$ bind. This is easy to verify when the problem has no constraints or only monotonicity constraints, which are by definition local (e.g. a single agent decision problem for choosing a strategy function in an auction). Another common case is when the integral is over private types, and $\mathbf{R}$ represents incentive compatibility constraints. In this case, the requirement is that local incentive compatibility is sufficient for global incentive compatibility, and this is a well studied problem. For example, Carroll (2012) presents sufficient conditions for a wide set of applications.

Second, if the state space $T$ is multidimensional, maintaining tractability requires stronger restrictions than the sufficiency of local constraints. The restriction to local constraints means that in the ordering of grid points over $T$, only consecutive constraints can bind. However, if the problem has a multidimensional integral over $T$, and is integrated from the inside to the out, the set of consecutive constraints would not be a sufficient set of constraints to ensure local incentive compatibility, which must hold along all paths. One solution for maintaining these constraints would be to expand $t_n$. For example, with a two dimensional type space, $t_n$ could be the set of all type grid points with the same x-coordinate. However, this causes the number of calculations to explode and solving this quickly becomes intractable. Although iterated approaches such as the one proposed in Section 3.2 can help, they omit some constraints and therefore result in upper bounds on profits. Alternatively, some problems require less constraints: for some cases it may be possible to show that under some ordering of types the consecutive constraints continue to imply incentive compatibility, or it may be possible to adopt an alternative formulation (e.g. in inverse strategies) that does not require local constraints.

Finally, if the full equilibrium algorithm is used (as opposed to the iterated equilibrium algorithm), there may be substantial advantages from using the inverse strategy representation. In many games this inverse strategy representation does not exist, as it relies on monotonicity of the strategy function, but it is satisfied in some important cases (e.g. many auction models). To see why this is useful for solving equilibrium, consider the auction example. At any price $p$, the opponents' threshold value uniquely pins down the win probability. This means that at any step $n$, we only need $x(t_{n-1})$ (rather than the entire function $x(t)$) of all bidders to solve for the equilibrium fixed point at step $n$. This is not true in the strategy space: for example, in an asymmetric auction, the win probability locally around $b$, depends on which types are choosing bids $b$, and this is described by $x(t)$ at an unknown type value $t$. Hence, the fixed point cannot be solved locally at $t$, and the efficiency gain from using the algorithm is largely lost.

## B.2 Discretization and Choice of Grids

In any problem, the grid densities for both the state and action functions will determine the computation times and accuracy. To help the user understand this trade-off, we show the sources of error, going from the original problem in its continuous version to the discrete approximation. We first consider the case of a one-dimensional integration and one-dimensional control functions, and then discuss how these results extend to multidimensional cases.

We proceed to show the approximation error by steps:

1. Suppose for ease of notation that the state is in the unit interval, $t \in [0, 1]$, and that the state distribution does not depend on $x$ so that the distribution function of $t$ reduces to $F(t)$. Furthermore, suppose that $F(t)$ is differentiable with $F'(t) = f(t)$. The the original problem can then be described as:

$$\max_{x(t)} \int_0^1 \Pi(x(t), t) f(t) dt.$$

2. We first focus on the state space, $T$, and decompose the problem by breaking it into $N$ discrete increments:

$$\max_{x(t)} \sum_n \int_{t_{n-1}}^{t_n} \Pi(x(t), t) f(t) dt.$$

3. We now let $x(t)$ be constant within each segment. Let $\tilde{x}_n$ be the value of $x_n$ such that $\Pi(x^*(t_n), t_n) = \Pi(\tilde{x}_n, t_n)$, where $x^*$ denotes the solution to the problem in step 1 or equivalently in step 2. By the mean value theorem, we have the following in each segment:

$$\begin{aligned} &\left| \int_{t_{n-1}}^{t_n} \Pi(x^*(t), t) f(t) dt - (t_n - t_{n-1}) \Pi(\tilde{x}_n, t_n) f(t_n) \right| \\ \leq &\frac{(t_n - t_{n-1})^2}{2} \sup_{t \in [t_{n-1}, t_n]} \frac{d(\Pi(x^*(t), t) f(t))}{dt}. \end{aligned}$$

We assume that the derivative is bounded by some constant $\Delta_D$, which depends on the specific problem. Note that we can define a constant $\Delta_F$ so that

$$\begin{aligned} &\left| \int_{t_{n-1}}^{t_n} \Pi(x^*(t), t) f(t) dt - \Pi(\tilde{x}_n, t_n)(F(t_n) - F(t_{n-1})) \right| \\ \leq &\left| \int_{t_{n-1}}^{t_n} \Pi(x^*(t), t) f(t) dt - (t_n - t_{n-1}) \Pi(\tilde{x}_n, t_n) f(t_n) \right| + \Delta_F. \end{aligned}$$

This implies that

$$\begin{aligned}
&\sum_n \Pi(\tilde{x}_n, t)(F(t_n) - F(t_{n-1})) \\
&\geq \max_{x(t)} \left\{ \sum_n \int_{t_{n-1}}^{t_n} \Pi_i(x(t), t) f(t) dt \right\} - N \frac{(t_n - t_{n-1})^2}{2} \Delta_D - \Delta_F.
\end{aligned}$$

4. From the optimization of a constrained versus an unconstrained problem, we obtain

$$\begin{aligned}
\sum_n \Pi(\tilde{x}_n, t)(F(t_n) - F(t_{n-1})) &\leq \max_{\hat{x}_1,.,,\hat{x}_N} \sum_n \Pi(\hat{x}_n, t)(F(t_n) - F(t_{n-1})) \\
&\leq \max_{x(t)} \sum_n \int_{t_{n-1}}^{t_n} \Pi(x(t), t) f(t) dt.
\end{aligned}$$

Thus, the profits from the approximated problem are smaller than the true profits over all possible functions $x(t)$, but bigger than the profit function evaluated at the optimal solution with profits corresponding at end points. This means

$$\begin{aligned}
&\left| \max_{\hat{x}_1,.,,\hat{x}_N} \sum_n \Pi(\hat{x}_n, t)(F(t_n) - F(t_{n-1})) - \max_{x(t)} \sum_n \int_{t_{n-1}}^{t_n} \Pi(x(t), t) f(t) \right| \\
\leq & N \frac{(t_n - t_{n-1})^2}{2} \Delta_D + \Delta_F.
\end{aligned}$$

Hence, this bounds the error from maximizing the approximated profits rather than the true profits.

5. We now focus on the action space. Suppose the action space is discrete: $\hat{x}_n \in (y_1, .., y_K)$, and let $\hat{x}_n^*$ be the nearest grid point to approximating the best profits at the right hand grid point i.e. $\hat{x}_n^* = \text{argmin}_{\hat{x}_n \in Y} |\Pi(x^*(t_n), t_n) - \Pi(\hat{x}_n, t_n)|$. Then

$$\begin{aligned}
&\left| \max_{x_1,.,,x_N} \sum_n \Pi(x_n, t)(F(t_n) - F(t_{n-1})) - \max_{x(t)} \sum_n \int_{t_{n-1}}^{t_n} \Pi(x(t), t) f(t) dt \right| \\
\leq & \left| \sum_n \Pi(\hat{x}_n^*, t)(F(t_n) - F(t_{n-1})) - \max_{x(t)} \sum_n \int_{t_{n-1}}^{t_n} \Pi(x(t), t) f(t) dt \right| \\
& + N \times \max_n (|\Pi(x^*(t_n), t_n) - \Pi(\hat{x}_n^*, t_n)|)(t_n - t_{n-1}) \\
\leq & N \frac{(t_n - t_{n-1})^2}{2} \Delta_D + N(t_n - t_{n-1}) \Delta_M + \Delta_F,
\end{aligned}$$

where $\Delta_M$ is a constant that bounds $\max_n(|\Pi(x^*(t_n), t_n) - \Pi(\hat{x}_n^*, t_n)|)$. Finally, note that as above, $\max_{x_1,.,,x_N, \in Y} \sum_n (F(t_n) - F(t_{n-1}))\Pi(x_n, t)$ is weakly greater than the sum evaluating $\Pi$ at the best grid-approximation to the right end points, and so the difference between those profits and the true (unconstrained) maximum solution is bounded by $N \frac{(t_n - t_{n-1})^2}{2} \Delta_D + N(t_n - t_{n-1}) \Delta_M + \Delta_F$.

6. Now we consider solving $\max_{x_1,.,.,x_N} \sum_n \Pi(x_n,t)(F(t_n)-F(t_{n-1}))$. This is the basic discrete problem that is input into the sequential formulation. We can rewrite the sum using step-$n$ value functions, with any $V_n(x) = \sum_{n'=1}^{n} \Pi(x_{n'},t)(F(t_{n'}) - F(t_{n'-1}))$ to obtain the sequential version.

As long as the slope of the profit function is not too steep, the errors from discretization over the dimension of integration are quite small: it is bounded by $N\frac{(t_n-t_{n-1})^2}{2}\Delta_D$, and even a fairly coarse set of grid points should provide a reasonable approximation. To avoid over-stating profits when working with a problem with constraints, it may be important to increase the grid density in this dimension. The errors in discretization of the action space grow instead at $N(t_n - t_{n-1})\Delta_M$. In the simple example, these increase compute time at $K^2$, so this increase is more costly. In balance, in our applications we found that this grid should be more dense than the dimension of integration.

## B.3 Practical Issues for Iterated Equilibrium Solution

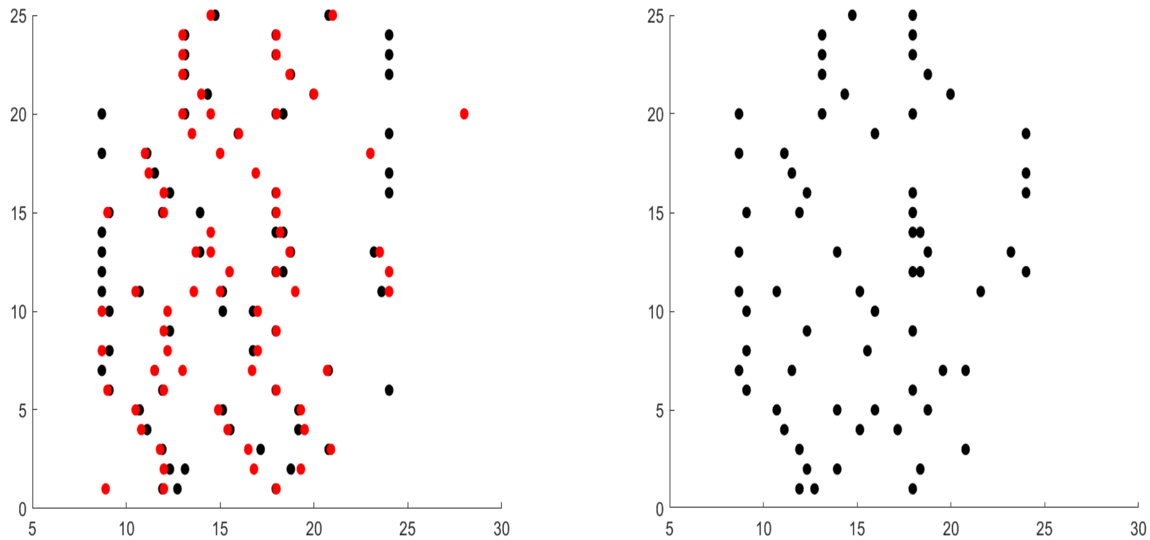
Since the iterated algorithm might not converge, and we have less guarantees of its properties, use of this algorithm requires additional attention in practice. There are two important considerations that apply to this algorithm, but are not otherwise relevant with our method: multiple solutions and smoothing to avoid cycling.

First, on the issue of multiplicity, we have little useful guidance in the iterative equilibrium algorithm. The user can try multiple start points as with other approaches. However, if multiplicity is a primary concern, perhaps the full equilibrium algorithm should be adopted instead of the iterated equilibrium algorithm, as the former can capture multiplicity, as discussed in the text.

Second, on the issue of avoiding cycles in the iterative equilibrium algorithm, we have had better results. Given the discrete approximations, performing best response iteration directly tends to lead to cycling behavior and the strategies fail to converge. Depending on the example, a variety of approaches are possible to smooth the results and avoid the cycles. For example, in the nonlinear pricing problem, we update the utility offered to each type by the best opposing option by taking a weighted average of the current best utility and the best utility calculated in the previous iteration. Thus, we update in the direction suggested by the algorithm but we dampen the step size of the update. In the auction example, we calculate the win probabilities by approximating the distribution of bids made by each bidder at grid points defined by the median prices used at each step in their current demand function. We then map this to win probabilities for each bidder at each price level, and on each iteration we update the distribution using a weighted average of the old and new win probability distributions. Again, this takes a step towards the new win probability distribution, implied by the updated strategies, but the size of the step is dampened to help avoid cycling.

# Appendix C Additional Figures and Tables

Figure A.1: Counterfactual Box Sizes




(A) Quantities Computed

(B) Quantities Cost Change

Red dots are actual menu points, black model predictions. X-axis is box-size in oz. and y-axis lists the brands.

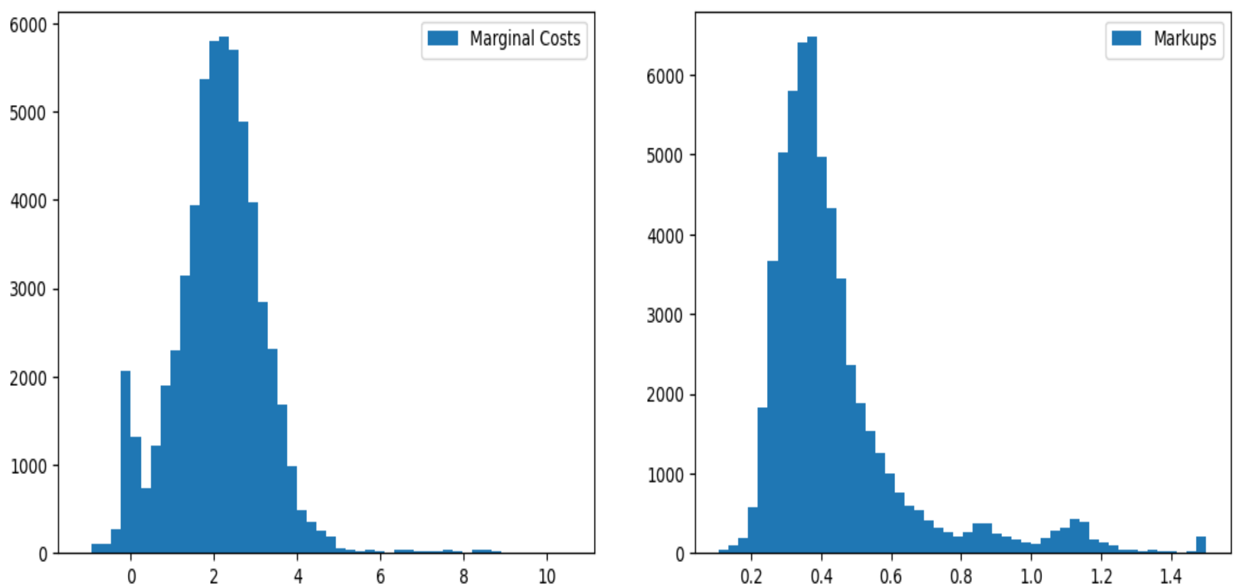
# Appendix D Estimation Details

A key decision is the share of inside good/outside good. we set this following the average share from Backus et al. (2021).

We calculate the micro moments from micro data, including: the covariance between package size and income, conditional on purchasing something, the covariance between size and kids, conditional on purchasing something, the average income conditional on purchasing something and the average kids conditional on purchasing something.

Marginal costs and markups implied are presented in Figure A.2.

FIGURE A.2: MARGINAL COSTS AND MARKUPS




(A)MC (B) Markups

NOTE: THESE ARE THE ESTIMATED MARGINAL COSTS AND MARKUPS. FOR THE FIGURE, THE MARKUPS ARE TRUNCATED AT 1.5, PRODUCING THE SMALL SPIKE.

## References


Armantier, O., J.-P. Florens, and J.-F. Richard (2008): "Approximation of Nash equilibria in Bayesian games," *Journal of Applied Econometrics*, 23, 965–981.

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.

Borkovsky, R. N., U. Doraszelski, and Y. Kryukov (2010): "A User's Guide to Solving Dynamic Stochastic Games Using the Homotopy Method," *Operations Research*, 58, 1116–1132.

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.

Crawford, G. S., O. Shcherbakov, and M. Shum (2019): "Quality Overprovision in Cable Television Markets," *American Economic Review*, 109, 956–995.

Crawford, G. S. and M. Shum (2007): "Monopoly Quality Degradation and Regulation in Cable Television," *The Journal of Law and Economics*, 50, 181–219.

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

Dharanan, G. and A. Ellis (2024): "Asymmetric auctions: Perturbations, $\epsilon$- 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," *Working paper.*

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

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 (2015): "Asymmetric Auctions with More Than Two Bidders," *Working paper.*

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.

Judd, K. L. (1998): *Numerical Methods in Economics*, vol. 1 of *MIT Press Books*, The MIT Press.

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.

Matthews, S. and J. Moore (1987): "Monopoly Provision of Quality and Warranties: An Exploration in the Theory of Multidimensional Screening," *Econometrica*, 55, 441–467.

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.

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. 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.

Rosa, B. V. (2019): "Resident Bid Preference, Affiliation, and Procurement Competition: Evidence from New Mexico," *Journal of Industrial Economics*, 67, 161–208.

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.

Wilson, R. (1997): *Nonlinear Pricing*, Oxford University Press.

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