RAND Journal of Economics
Vol. 43, No. 2, Summer 2012
pp. 253–282

An approximate dynamic programming approach to solving dynamic oligopoly models

Vivek Farias*

Denis Saure**

and

Gabriel Y. Weintraub***

In this article, we introduce a new method to approximate Markov perfect equilibrium in large-scale Ericson and Pakes (1995)-style dynamic oligopoly models that are not amenable to exact solution due to the curse of dimensionality. The method is based on an algorithm that iterates an approximate best response operator using an approximate dynamic programming approach. The method, based on mathematical programming, approximates the value function with a linear combination of basis functions. We provide results that lend theoretical support to our approach. We introduce a rich yet tractable set of basis functions, and test our method on important classes of models. Our results suggest that the approach we propose significantly expands the set of dynamic oligopoly models that can be analyzed computationally.

1. Introduction

■ In a pioneering article, Ericson and Pakes (1995) (hereafter, EP) introduced a framework to model a dynamic industry with heterogeneous firms. The stated goal of that work was to facilitate empirical research analyzing the effects of policy and environmental changes on such things as market structure and consumer welfare in different industries. Due to the importance of dynamics in determining policy outcomes, and also because the EP model has proved to be quite adaptable and broadly applicable, the model has lent itself to many applications (see Doraszelski

*Massachusetts Institute of Technology; vivekf@mit.edu.
**University of Pittsburgh; dsaure@pitt.edu.
***Columbia University; gweintraub@columbia.edu.

We have had very helpful conversations with Lanier Benkard, Allan Collard-Wexler, Uli Doraszelski, Ariel Pakes, Carlos Santos, and Ben Van Roy, as well as seminar participants at Columbia Business School, IIOC, Informs, MSOM Conference, the Econometric Society Summer Meeting, and the NYU-Kansas City Fed Workshop on Computational Economics. We thank the editor, Phil Haile, and two anonymous referees for suggestions that greatly improved the article. The research of Farias was supported, in part, by the Solomon Buchsbaum Research Fund.

and Pakes, 2007 for an excellent survey).[1] With the introduction of new estimation methods (see Pesendorfer and Schmidt-Dengler, 2008; Bajari, Benkard, and Levin, 2007; Pakes, Ostrovsky, and Berry, 2007; Aguirregabiria and Mira, 2007), this has also become an active area for empirical research.

There remain, however, some substantial hurdles in the application of EP-style models in practice. Because EP-style models are typically analytically intractable, their solution involves numerically computing their Markov perfect equilibria (MPE) (e.g., Pakes and McGuire, 1994). The practical applicability of EP-style models is severely limited by the "curse of dimensionality" from which this computation suffers. Note that even if it is possible to estimate the model parameters without computing an equilibrium, as in the articles listed above, equilibrium computation is still required to analyze the effects of a policy or other environmental change. Methods that accelerate these equilibrium computations have been proposed (Judd, 1998; Pakes and McGuire, 2001; Doraszelski and Judd, 2010). However, in practice, computational concerns have typically limited the analysis to industries with just a few firms (say, two to six), far fewer than the real-world industries at which the analysis is directed. Such limitations have made it difficult to construct realistic empirical models.

Thus motivated, we introduce in this article a new method to approximate MPE in EP-style dynamic oligopoly models based on approximate dynamic programming. Our method opens the door to solving problems that, given currently available methods, have to this point been infeasible. In particular, our method offers a viable means to approximating MPE in dynamic oligopoly models with large numbers of firms, enabling, for example, the execution of counterfactual experiments. We believe this substantially enhances the applicability of EP-style models.

In an EP-style model, each firm is distinguished by an *individual state* at every point in time. The value of the state could represent a measure of product quality, current productivity level, or capacity. The *industry state* is a vector encoding the number of firms with each possible value of the individual state variable. Assuming its competitors follow a prescribed strategy, a given firm must, at each point in time, select an action (e.g., an investment level) to maximize its expected discounted profits; its subsequent state is determined by its current individual state, its chosen action, and a random shock. The selected action will depend in general on the firm's individual state and the industry state. Even if firms were restricted to symmetric strategies, the computation entailed in selecting such an action quickly becomes infeasible as the number of firms and individual states grows. For example, in a model with 30 firms and 20 individual states, more than two million gigabytes would be required just to store a strategy function. This renders commonly used dynamic programming algorithms to compute MPE infeasible in many problems of practical interest.

The first main contribution of the article is to introduce a *tractable* algorithm to approximate MPE in large-scale EP-style dynamic oligopoly models. Our approach is based on an algorithm that iterates an "approximate" best response operator. In short, the value function is approximated by a linear combination of basis functions, and in each iteration we compute an approximation to the best response value function via the "approximate linear programming" approach (de Farias and Van Roy, 2003, 2004). We repeat this step until no more progress can be made. Our method can be applied to a general class of dynamic oligopoly models, and we numerically test our method on important classes of EP-style models. Our algorithm runs on the order of minutes to hours on a modern workstation, even in models with tens of firms and tens of individual states per firm.

Our scheme relies on approximating the best response value function with a linear combination of basis functions. The set of basis functions is an input for our algorithm, and choosing a "good" set of basis functions (which we also refer to as an approximation architecture)

[1] Indeed, recent work has applied the framework to studying problems as diverse as advertising, auctions, collusion, consumer learning, environmental policy, firm mergers, industry dynamics, limit order markets, network externalities, and R&D investment.

is a problem-specific task. It requires understanding what features of the state may have the largest impact on the value function and optimal strategy, and a fair amount of trial and error. We discuss how numerical experiments and economic intuition can help in the process of selecting good basis functions. Based on this, for the class of models we study in our computational experiments, we propose using a rich, but tractable, approximation architecture that captures a natural "moment"-based approximation architecture. With this set of basis functions and a suitable version of our approximate best response algorithm, we explore the problem of approximating MPE across various problem regimes.

More specifically, we provide an extensive computational demonstration of our method on two classes of EP-style models: (i) a quality ladder model similar to Pakes and McGuire (1994); and (ii) a capacity competition model motivated by Besanko and Doraszelski (2004). Similar models have been previously used as a test bed for new methods to compute and approximate MPE (Doraszelski and Judd, 2010; Weintraub, Benkard and Van Roy, 2010). To assess the accuracy of our approximation, we compare the candidate equilibrium strategy produced by the approach to computable benchmarks. First, in models with relatively few firms and few individual states, we can compute MPE exactly. We show that in these models our method provides accurate approximations to MPE with substantially less computational effort.

Next, we examine industries with a large number of firms and use "oblivious equilibrium" (henceforth, OE) introduced by Weintraub, Benkard and Van Roy (2008) as a benchmark. OE is a simple to compute equilibrium concept and provides valid approximations to MPE in several EP-style models with large numbers of firms. We compare the candidate equilibrium strategy produced by our approach to OE in parameter regimes where OE can be shown to be a good approximation to MPE. Here too we show that our candidate equilibrium strategy is close to OE and hence to MPE.

Our results suggest that our chosen approximation architecture together with our algorithm provide accurate approximations to MPE in the two regimes described above. Moreover, our results show that a relatively compact set of basis functions that captures few features of the industry state allows approximating MPE accurately.

Outside of the regimes above, there is a large "intermediate" regime for which no benchmarks are available. In particular, this regime includes problems that are too large to be solved exactly and for which OE is not known to be a good approximation to MPE. Examples of problems in this regime are many large industries (say, with tens of firms) in which the few largest firms hold a significant market share. This is a commonly observed market structure in real-world industries. In these intermediate regimes our scheme is convergent, but it is difficult to make comparisons to alternative methods to gauge the validity of our approximations because no such alternatives are available. Nonetheless, the experience with the two aforementioned regimes suggests that our approximation architecture should also be capable of capturing the true value function in the intermediate regime and that our method will produce effective approximations to MPE here as well. We believe our method offers the first viable approach to approximating MPE in these intermediate regimes, significantly expanding the range of industries that can be analyzed computationally.

Finally, another important contribution of our work is a series of results that give theoretical support to our approximation. These results are valid for a general class of dynamic oligopoly models. In particular, we propose a simple, easily computable convergence criterion for our algorithm that lends itself to a theoretical guarantee of the following flavor. Assume that our iterative scheme converges. Further, assume that a good approximation to the value function corresponding to our candidate equilibrium strategy is within the span of our chosen basis functions. Then, upon convergence, we are guaranteed to have computed a good approximation to an MPE. It is worth noting that such guarantees are typically not available for other means of approximating best responses such as approximate value iteration-based methods (Bertsekas and Tsitsiklis, 1996). We believe this is an important advantage of our approximate mathematical programming approach.

The article is organized as follows. Section 2 describes related literature. In Section 3, we introduce our dynamic oligopoly model. In Section 4, we introduce our equilibrium concept and discuss its computation. In Section 5, we describe the main elements of our approximate dynamic programming approach and discuss value function approximation; this discussion remains at a relatively conceptual level. In Section 6, we report results from computational experiments. In Section 7, we provide conclusions and discuss extensions of our work. In addition, in the online companion to this article, we provide several appendices with important content. Web Appendix A develops in detail the linear programming formulation. Web Appendix B provides a "guide for practitioners" of our algorithm at a level of detail of interest to readers implementing the approach. Web Appendices C and D provide specifics about the theory that give support to our approach in terms of approximation guarantees. In addition, the code used in our numerical experiments is publicly available on the authors' webpages.

2. Related literature

■ Our work extends the approximate linear programming approach to dynamic programming (de Farias and Van Roy, 2003, 2004) to consider a dynamic game setting. The extension requires dealing with new computational challenges that inherently arise in the context of a best response algorithm. We also extend the theory to obtain useful guarantees in this context, where we are interested in approximating an equilibrium as oppose to a single-agent optimization problem.

As we have discussed above, our work is also related to Weintraub, Benkard, and Van Roy (2008, 2010). Like them, we consider algorithms that can efficiently deal with large numbers of firms, but aim to compute an approximation rather than an exact MPE and provide bounds for the error. Our work complements OE, in that we can potentially approximate MPE in situations where OE is not known to be a good approximation while continuing to provide good approximations to MPE where OE does indeed serve as a good approximation, albeit at a higher computational cost.

Our work is also related to Pakes and McGuire (2001), who introduced a stochastic algorithm that uses simulation to sample and concentrate the computational effort on relevant states. Judd (1998) discusses value function approximation techniques for dynamic programs with continuous state spaces. Doraszelski (2003), among others, has applied the latter method for dynamic games with a low-dimensional continuous state space. Trick and Zin (1993, 1997) use the linear programming approach in two-dimensional problems that arise in macroeconomics. As far as we know, our article is the first to combine a simulation scheme to sample relevant states (a procedure inherent to the approximate mathematical programming approach) together with value function approximation to solve highly dimensional dynamic oligopoly models.

Pakes and McGuire (1994) suggest using value function approximation for EP-style models within a value iteration algorithm, but report serious convergence problems. In their handbook chapter, Doraszelski and Pakes (2007) argue that value function approximation may provide a viable alternative to solving large-scale dynamic stochastic games, but that further developments are needed. We believe this article provides one path toward those developments.

Finally, we defer more specific discussion of the comparison of our approach to some commonly used alternatives to Section 5, after introducing our method.

3. A dynamic oligopoly model

■ We consider a variation of the industry model in Weintraub, Benkard and Van Roy (2008) (which in turn is close in spirit to Ericson and Pakes, 1995) where firms compete in a single-good market and the industry evolves over discrete time periods and an infinite horizon. At the end of the section, we describe two specific versions of the model that we will use as a test bed for our methods: a quality ladder model similar to Pakes and McGuire (1994) and a capacity model based on Besanko and Doraszelski (2004).

□ **Model and notation.** We index time periods with non negative integers $t \in \mathbb{N}$ ($\mathbb{N} = \{0, 1, 2, \ldots\}$). Each incumbent firm is assigned a unique positive integer-valued index.

State space. Firm heterogeneity is reflected through firm states. Firm states might reflect quality level, productivity, capacity, the size of its consumer network, or any other aspect of the firm that affects its profits. At time *t*, the individual state of firm *i* is denoted by $x_{it} \in \mathcal{X} = \{0, 1, 2, \ldots, \overline{x}\}$. The integer number $\overline{x}$ is an upper bound on firms' individual states. We define the *industry state* s_t to be a vector over individual states that specifies, for each firm state $x \in \mathcal{X}$, the number of incumbent firms at x in period *t*. We define the state space $\mathcal{S} = \{s \in \mathbb{N}^{|\mathcal{X}|} \mid \sum_{x=0}^{\overline{x}} s(x) \leq N\}$. Note that because we will focus on symmetric and anonymous equilibrium strategies in the sense of Doraszelski and Pakes (2007), we can restrict the state space so that the identity of firms does not matter. The integer number *N* represents the maximum number of incumbent firms that the industry can accommodate at every point in time. We let n_t be the number of incumbent firms at time period *t*, that is, $n_t = \sum_{x=0}^{\overline{x}} s_t(x)$.

Single-period profit function. Each incumbent firm earns profits on a spot market. For firm *i*, its single-period expected profits $\pi(x_{it}, s_t)$ depend on its individual state $x_{it} \in \mathcal{X}$ and the industry state $s_t \in \mathcal{S}$. We assume profits are bounded.

Exit process. The model allows for entry and exit. In each period, each incumbent firm *i* observes a positive real-valued sell-off value κ_{it} that is private information to the firm. If the sell-off value exceeds the value of continuing in the industry then the firm may choose to exit, in which case it earns the sell-off value and then ceases operations permanently. We assume the random variables $\{\kappa_{it} \mid t \geq 0, i \geq 1\}$ are i.i.d. and have a well-defined density function with support on the positive real line and finite moments.

Investment dynamics. Firms that decide to remain in the industry can invest (at a cost of *d* per unit) to improve their individual states. If a firm invests $\iota_{it} \in \mathbb{R}_+$, then the firm's state at time $t + 1$ is given by

$$x_{i,t+1} = x_{it} + w(x_{it}, \iota_{it}, \zeta_{i,t+1}),$$

where the function w captures the impact of investment on individual state and $\zeta_{i,t+1}$ reflects idiosyncratic uncertainty in the outcome of investment. We assume the random variables $\{\zeta_{it} \mid t \geq 0, i \geq 1\}$ are i.i.d. and independent of $\{\kappa_{it} \mid t \geq 0, i \geq 1\}$. Uncertainty may arise, for example, due to the risk associated with a research and development endeavor or a marketing campaign. To simplify notation, we do not consider an industry-wide shock to investment dynamics, but our methods could easily accommodate one.

We assume investment is bounded by some constant $\bar{\iota}$. We also assume that the impact of investment on transition probabilities is continuous. Finally, we assume the transitions generated by $w(x, \iota, \zeta)$ are unique investment choice admissible. This last assumption is introduced by Doraszelski and Satterthwaite (2010) and ensures a unique solution to the firms' investment decision problem. In particular, it ensures the firms' investment decision problem is strictly concave or that the unique maximizer is a corner solution. The assumption is used to guarantee existence of an equilibrium in pure strategies, and is satisfied by many of the commonly used specifications in the literature.

Entry process. We consider an entry process similar to the one in Doraszelski and Pakes (2007). At time period *t*, there are $N - n_t$ potential entrants, ensuring that the maximum number of incumbent firms that the industry can accommodate is *N*.[2] Each potential entrant is assigned a unique positive integer-valued index. In each time period, each potential entrant *i* observes a positive real-valued entry cost ϕ_{it} that is private information to the firm. We assume the random variables $\{\phi_{it} \mid t \geq 0 i \geq 1\}$ are i.i.d. and independent of $\{\kappa_{it}, \zeta_{it} \mid t \geq 0 i \geq 1\}$, and have a well-defined density function with support on the positive real line and finite moments. If the

[2] We assume $n_0 \leq N$.

entry cost is below the expected value of entering the industry, then the firm will choose to enter.

Potential entrants make entry decisions simultaneously. Entrants do not earn profits in the period they decide to enter. They appear in the following period at state $x^e \in \mathcal{X}$ and can earn profits thereafter.[3] As is common in this literature and to simplify the analysis, we assume potential entrants are short lived and do not consider the option value of delaying entry. Potential entrants that do not enter the industry disappear, and a new generation of potential entrants is created next period.

Timing of events. In each period, events occur in the following order: (i) each incumbent firm observes its sell-off value and then makes exit and investment decisions; (ii) each potential entrant observes its entry cost and makes entry decisions; (iii) incumbent firms compete in the spot market and receive profits; (iv) exiting firms exit and receive their sell-off values; and (v) investment shock outcomes are determined, new entrants enter, and the industry takes on a new state s_{t+1}.

Firms' objective. Firms aim to maximize expected net present value: the interest rate is assumed to be positive and constant over time, resulting in a constant discount factor of $\beta \in (0, 1)$ per time period.

□ **Specific models.** The model described above is general enough to encompass numerous applied problems in economics. To study any particular problem it is necessary to further specify the primitives of the model. In this section, we briefly describe two specifications that we consider in our numerical experiments. Full details of the model primitives and parameters are provided in Section 6, later in the article.

For implementation simplicity, we consider exponentially distributed random variables to model both the sell-off value and the entry cost. Following Pakes and McGuire (1994), a firm that invests a quantity ι is successful with probability $(\frac{b\iota}{1+b\iota})$, in which case the individual state increases by one level. The firm's individual state depreciates by one state with probability δ, independently each period. Independent of everything else, every firm has a probability γ of increasing its state by one level.[4] Hence, a firm can increase its state even in the absence of investment. If the appreciation shock is unsuccessful, then the transitions are determined by the investment and depreciation processes.

Note that in most applications the profit function would not be specified directly, but would instead result from a deeper set of primitives that specify a demand function, a cost function, and a static equilibrium concept. Next, we specify two models that we will use in our computational experiments. The entry, exit, and investment processes are kept the same for both of these models.

Profit function: quality ladder model. Similarly to Pakes and McGuire (1994), we consider an industry with differentiated products, where each firm's state variable represents the quality of its product. Given quality levels and prices, demand is described by a standard logit model. All firms share the same constant marginal cost of production. Profits at each period are determined by the unique Nash equilibrium of the pricing game among firms.

Profit function: capacity competition model. This model is based on the quantity competition version of Besanko and Doraszelski (2004). We consider an industry with homogeneous products, where each firm's state variable determines its production capacity. All firms share the same constant marginal cost of production. There is a linear demand function. At each period, firms

[3] It is straightforward to generalize the model by assuming that entrants can also invest to improve their initial state.

[4] In our experiments, we eventually consider both settings where $\gamma = 0$ and $\gamma > 0$. As we will explain in more detail in Web Appendix B.1, assuming a (small) positive value of γ encourages "exploration" in larger parts of the state space in the course of our algorithm. In our experience, we observed that sometimes this improved the approximation.

compete in a capacity-constrained quantity setting game. Profits are determined by the unique Nash equilibrium of this game.

4. Equilibrium

■ In this section, we introduce our notion of equilibrium and present a best response algorithm to compute it. Then, we argue that solving for a best response is infeasible for many problems of practical interest. This motivates our approach of finding *approximate* best responses at every step instead, using approximate dynamic programming.

□ **Markov perfect equilibrium.** As a model of industry behavior, we focus on pure-strategy MPE, in the sense of Maskin and Tirole (1988). We further assume that equilibrium is symmetric, such that all firms use a common stationary investment/exit strategy. In particular, there is a function ι such that at each time t, each incumbent firm i invests an amount $\iota_{it} = \iota(x_{it}, s_t)$. Similarly, each firm follows an exit strategy that takes the form of a cutoff rule: there is a real-valued function ρ such that an incumbent firm i exits at time t if and only if $\kappa_{it} > \rho(x_{it}, s_t)$. Weintraub, Benkard, and Van Roy (2008) show that there always exists an optimal exit strategy of this form even among very general classes of exit strategies. Let $\mathcal{Y} = \{(x, s) \in \mathcal{X} \times \mathcal{S} : s(x) > 0\}$[5]. Let $\mathcal{M}$ denote the set of exit/investment strategies such that an element $\mu \in \mathcal{M}$ is a set of functions $\mu = (\iota, \rho)$, where $\iota : \mathcal{Y} \rightarrow \mathbb{R}_+$ is an investment strategy and $\rho : \mathcal{Y} \rightarrow \mathbb{R}$ is an exit strategy.

Similarly, each potential entrant follows an entry strategy that takes the form of a cutoff rule: there is a real-valued function λ such that a potential entrant i enters at time t if and only if $\phi_{it} < \lambda(s_t)$. It is simple to show that there always exists an optimal entry strategy of this form even among very general classes of entry strategies (see Doraszelski and Satterthwaite, 2010). We denote the set of entry strategies by Λ, where an element of Λ is a function $\lambda : \mathcal{S}^e \rightarrow \mathbb{R}$ and $\mathcal{S}^e = \{s \in \mathcal{S} : \sum_{x=0}^{\overline{x}} s(x) < N\}$. Note that $\mathcal{S}^e$ is the set of industry states with fewer than N firms, that is, with a positive number of potential entrants.

We define the value function $V_{\mu,\lambda}^{\mu'}(x, s)$ to be the expected discounted value of profits for a firm at state x when the industry state is s, given that its competitors each follows a common strategy $\mu \in \mathcal{M}$, the entry strategy is $\lambda \in \Lambda$, and the firm itself follows strategy $\mu' \in \mathcal{M}$. In particular,

$$V_{\mu,\lambda}^{\mu'}(x, s) = E_{\mu,\lambda}^{\mu'}\left[\sum_{k=t}^{\tau_i} \beta^{k-t}(\pi(x_{ik}, s_k) - d\iota_{ik}) + \beta^{\tau_i - t}\kappa_{i,\tau_i} \middle| x_{it} = x, s_t = s\right],$$

where i is taken to be the index of a firm at individual state x at time t, τ_i is a random variable representing the time at which firm i exits the industry, and the superscript and subscripts of the expectation indicate the strategy followed by firm i, the strategy followed by its competitors, and the entry strategy. In an abuse of notation, we will use the shorthand, $V_{\mu,\lambda}(x, s) \equiv V_{\mu,\lambda}^{\mu}(x, s)$, to refer to the expected discounted value of profits when firm i follows the same strategy μ as its competitors. An equilibrium in our model comprises an investment/exit strategy $\mu = (\iota, \rho) \in \mathcal{M}$ and an entry strategy $\lambda \in \Lambda$ that satisfy the following conditions.

(i) Incumbent firm strategies represent an MPE:

$$\sup_{\mu' \in \mathcal{M}} V_{\mu,\lambda}^{\mu'}(x, s) = V_{\mu,\lambda}(x, s) , \quad \forall (x, s) \in \mathcal{Y}. \tag{1}$$

(ii) For all states with a positive number of potential entrants, the cutoff entry value is equal to the expected discounted value of profits of entering the industry[6]:

$$\lambda(s) = \beta E_{\mu,\lambda}[V_{\mu,\lambda}(x^e, s_{t+1}) \,|\, s_t = s], \ \forall s \in \mathcal{S}^e. \tag{2}$$

[5] By $s(x)$, we understand the xth component of s.

[6] Hence, potential entrants enter if the expected discounted profits of doing so is positive. Throughout the article it is implicit that the industry state at time period $t + 1$, s_{t+1}, includes the entering firm in state x^e whenever we write (x^e, s_{t+1}).

Standard dynamic programming arguments establish that the supremum in part (i) of the definition above can always be attained simultaneously for all x and s by a common strategy μ'. Doraszelski and Satterthwaite (2010) establish existence of an equilibrium in pure strategies for this model. With respect to uniqueness, in general we presume that our model may have multiple equilibria. Doraszelski and Satterthwaite (2010) also provide an example of multiple equilibria in a closely related model.[7]

□ **Computation of MPE.** Although there are different approaches to computing MPE, a natural method is to iterate a best response operator. Dynamic programming algorithms can be used to optimize firms' strategies at each step. Stationary points of such iterations are MPEs. With this motivation, we define a best response operator. For all $\mu \in \mathcal{M}$ and $\lambda \in \Lambda$, we denote the best response investment/exit strategy as $\mu^*_{\mu,\lambda}$. The best response investment/exit strategy solves $\sup_{\mu' \in \mathcal{M}} V^{\mu'}_{\mu,\lambda} = V^{\mu^*_{\mu,\lambda}}_{\mu,\lambda}$, where the supremum is attained pointwise. To simplify notation, we will usually denote the best response to (μ, λ) by μ^*. We also define the best response value function as $V^*_{\mu,\lambda} = V^{\mu^*}_{\mu,\lambda}$. Now, for all $\mu \in \mathcal{M}$ and $\lambda \in \Lambda$, we define the best response operator $\mathcal{BR} : \mathcal{M} \times \Lambda \to \mathcal{M} \times \Lambda$ according to $\mathcal{BR}(\mu, \lambda) = (\mathcal{BR}_1(\mu, \lambda), \mathcal{BR}_2(\mu, \lambda))$, where

$$\mathcal{BR}_1(\mu, \lambda) = \mu^*_{\mu,\lambda},$$

$$\mathcal{BR}_2(\mu, \lambda)(s) = \beta E_{\mu,\lambda}\left[V^*_{\mu,\lambda}(x^e, s_{t+1}) \middle| s_t = s \right], \quad \forall s \in \mathcal{S}^e.$$

A fixed point of the operator $\mathcal{BR}$ is an MPE. Starting from an arbitrary strategy $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, we introduce the following iterative best response algorithm.[8]

Algorithm 1. Best response algorithm for MPE

```
1: $\mu_0 := \mu$ and $\lambda_0 := \lambda$
2: $i := 0$
3: repeat
4:     $(\mu_{i+1}, \lambda_{i+1}) = \mathcal{BR}(\mu_i, \lambda_i)$
5:     $\Delta := \|(\mu_{i+1}, \lambda_{i+1}) - (\mu_i, \lambda_i)\|_\infty$
6:     $i := i + 1$
7: until $\Delta < \epsilon$
```

If the termination condition is satisfied with $\epsilon = 0$, we have an MPE. Small values of ϵ allow for small errors associated with limitations of numerical precision.

Step (4) in the algorithm (i) updates the entry strategy and (ii) requires solving a dynamic programming problem to optimize incumbent firms' strategies. The latter is usually done by solving Bellman's equation with a dynamic programming algorithm (Bertsekas, 2001). The size of the state space of this problem is equal to

$$|\mathcal{X}| \binom{N + |\mathcal{X}| - 1}{N - 1}.$$

Therefore, methods that attempt to solve the dynamic program exactly are computationally infeasible for many applications, even for moderate sizes of $|\mathcal{X}|$ and N. For example, a model with 20 firms and 20 individual states has more than a thousand billion states. This motivates our alternative approach, which relaxes the requirement of finding a best response in step (4) of the algorithm and finds an approximate best response instead.

[7] We note, however, that using the (approximate) best response algorithm that we introduce below, we have not been able to find two different (approximate) MPEs for a given instance, even when starting from different initial conditions.

[8] Rust (1994) suggests a similar algorithm for exact computation of MPE.

5. Approximate dynamic programming

■ In this section, we first specialize Algorithm 1 by performing step (4) using the mathematical programming approach to dynamic programming that we describe in the next subsection. This method attempts to find a best response and, hence, it requires computation time and memory that grow proportionately with the number of relevant states, which, as mentioned above, is intractable in many applications. Then, we describe how to overcome the curse of dimensionality and simplify the computation following several steps. Each step is illustrated through examples. Notably, our approach reduces the dimensionality of the mathematical program significantly. In addition, it reduces the original non linear mathematical program to a linear program that is much easier to solve. In summary, step (4) of the algorithm finds an approximate best response by solving a *tractable* linear program, and in this way finds an approximation to MPE.

□ **Mathematical programming approach.** For some $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, consider the problem of finding a best response strategy $\mu^*_{\mu,\lambda}$. It is well known that the best response may be found computing a fixed point of the Bellman operator. We now construct the Bellman operator for our dynamic oligopoly model. Let us define for an arbitrary $\mu' \in \mathcal{M}$ the continuation value operator $C^{\mu'}_{\mu,\lambda}$ of a firm that decides to stay in the industry using strategy μ' according to

$$\left(C^{\mu'}_{\mu,\lambda} V\right)(x, s) = -d\iota'(x, s) + \beta E_{\mu,\lambda}[V(x_1, s_1) \,|\, x_0 = x, s_0 = s, \iota_0 = \iota'(x, s)], \quad \forall\, (x, s) \in \mathcal{Y},$$

where $V \in \mathbb{R}^{|\mathcal{Y}|}$ and (x_1, s_1) is random. Now, optimizing over the investment strategy, let us define the operator $C_{\mu,\lambda}$ according to

$$C_{\mu,\lambda} V(x, s) = \max_{\mu'(x,s):\iota'(x,s)\in[0,\ \bar{\iota}]} C^{\mu'}_{\mu,\lambda} V(x, s).$$

Define the operator $T^{\mu'}_{\mu,\lambda}$ according to

$$T^{\mu'}_{\mu,\lambda} V(x, s) = \pi(x, s) + \mathcal{P}[\kappa \geq \rho'(x, s)] E[\kappa \mid \kappa \geq \rho'(x, s)] + \mathcal{P}[\kappa < \rho'(x, s)] C^{\mu'}_{\mu,\lambda} V(x, s),$$

and the Bellman operator, $T_{\mu,\lambda}$, according to

$$T_{\mu,\lambda} V(x, s) = \pi(x, s) + E[\kappa \vee C_{\mu,\lambda} V(x, s)],$$

where $a \vee b = \max(a, b)$ and κ is drawn according to the sell-off value distribution presumed. Note that in the equation above, an incumbent firm optimizes over both its exit strategy and investment strategy. The best response to (μ, λ) may be found by computing a fixed point of this Bellman operator. In particular, it is simple to show $V^*_{\mu,\lambda}$ is the unique fixed point of this operator. The best response strategy, μ^*, may then be found as the strategy that achieves the maximum when applying the Bellman operator to the optimal value function (Bertsekas 2001). We call this strategy the "greedy" strategy with respect to $V^*_{\mu,\lambda}$. That is, a best response strategy μ^* may be identified as a strategy for which

$$T^{\mu^*}_{\mu,\lambda} V^*_{\mu,\lambda} = T_{\mu,\lambda} V^*_{\mu,\lambda},$$

where $V^*_{\mu,\lambda}$ is the unique fixed point of the Bellman operator $T_{\mu,\lambda}$. A solution to Bellman's equation may be obtained via a number of algorithms. One algorithm requires solving the following simple to state mathematical program:

$$\begin{aligned} \min \quad & c'V \\ \text{s.t.} \quad & (T_{\mu,\lambda} V)(x, s) \leq V(x, s), \quad \forall (x, s) \in \mathcal{Y}. \end{aligned} \tag{3}$$

It is a well-known fact that when c is a component-wise positive vector, the above program yields as its optimal solution the value function associated to a best response to (μ, λ), $V^*_{\mu,\lambda}$ (Bertsekas 2001).

If the state space is large, solving this mathematical program in step (4) of Algorithm 1 to find a best response poses a number of important challenges that we explain one by one in this section:

(i) Number of variables. The variable vector of the mathematical program is the value function V; its dimension is equal to the size of the state space. First, we show how to reduce the number of variables of the mathematical program using value function approximation.

(ii) Approximation error. If the state space is large, it is unlikely that the value function can be approximated uniformly well over the entire state space. Hence, we then discuss how the state relevance weight vector c plays the role of trading off approximation error across different states.

(iii) Number of constraints. The number of constraints of the mathematical program is equal to the size of the state space. We introduce a constraint sampling scheme to alleviate this difficulty.

(iv) Non linear program. Program (3) is, as stated, a non linear program. We observe, however, that it is a *convex* programming problem, and as such could potentially be efficiently solved given the previous steps, (i)–(iii); (see Boyd and Vandenberghe, 2004). Even so, we introduce a useful *linear* formulation that approximates the solution of this non linear, convex mathematical program that is even simpler to solve.

The above steps yield a tractable approach to computing an approximate best response. After explaining each of them, we integrate the approximate dynamic programming approach we have developed for the computation of an approximate best response in Algorithm 1 to yield a tractable approximation to MPE. We conclude the section by summarizing theory that provides support for our approach.

Before moving on, we digress to mention that many alternative methods for computing a good approximate best response to competitors' strategies do exist; the interested reader is referred to Bertsekas and Tsitsiklis (1996). In effect, all of these methods attempt to produce an approximate solution to the Bellman equation that uniquely determines a best response. A method popular in economic applications is the "collocation" method studied by Judd (1998). This method essentially enforces the Bellman equation at a few carefully chosen states. Our choice of the mathematical programming approach here is appealing for several reasons:

(i) The crucial computational step requires the solution of a linear (or, at worst, convex) program that is relatively easy to solve. In contrast, collocation methods, for example, typically require solving non linear systems of equations that are generally harder to solve.

(ii) The approach permits *approximation* and *performance* guarantees as previously mentioned. Other approaches based on collocation or value-iteration methods do not share these theoretical strengths. In fact, even a single-step best response computation based on the latter methods need not be convergent.

□ **Value function approximation.** Following de Farias and Van Roy (2003), we introduce value function approximation; we approximate the value function by a linear combination of basis functions. This reduces the number of variables in the program.

Assume we are given a set of "basis" functions $\Phi_i : \mathcal{Y} \to \mathbb{R}$, for $i = 1, 2, \ldots, K$. Let us denote by $\Phi \in \mathbb{R}^{|\mathcal{Y}| \times K}$ the matrix $[\Phi_1, \Phi_2, \ldots, \Phi_K]$. Given the difficulty in computing $V^*_{\mu,\lambda}$ exactly, we focus in this section on computing a set of weights $r \in \mathbb{R}^K$ for which Φr closely approximates $V^*_{\mu,\lambda}$. To that end, we consider the following program:

$$\begin{array}{ll} \min & c'\Phi r \\ \text{s.t.} & (T_{\mu,\lambda}\Phi r)(x,s) \leq (\Phi r)(x,s), \quad \forall (x,s) \in \mathcal{Y}. \end{array} \tag{4}$$

The above program attempts to find a good approximation to $V^*_{\mu,\lambda}$ within the linear span of the basis functions $\Phi_1, \Phi_2, \ldots, \Phi_K$. The idea is that if the basis functions are selected so that they can closely approximate the value function $V^*_{\mu,\lambda}$, then the program (4) should provide an effective approximation. By settling for an approximation to the optimal value function, we have reduced our problem to the solution of a mathematical program with a potentially small number of variables (K).

Given a good approximation to $V^*_{\mu,\lambda}$, namely $\Phi r_{\mu,\lambda}$, with $r_{\mu,\lambda}$ a solution of the mathematical program above, one may consider using as a proxy for the best response strategy the greedy strategy with respect to $\Phi r_{\mu,\lambda}$, namely, a strategy $\tilde{\mu}$ satisfying

$$T^{\tilde{\mu}}_{\mu,\lambda}\Phi r_{\mu,\lambda} = T_{\mu,\lambda}\Phi r_{\mu,\lambda}.$$

Provided $\Phi r_{\mu,\lambda}$ is a good approximation to $V^*_{\mu,\lambda}$, the expected discounted profits associated with using strategy $\tilde{\mu}$ in response to competitors that use strategy μ and entrants that use strategy λ should also be close to $V^*_{\mu,\lambda}$. We formalize these notions in web Appendices C.1 and C2.

We note that our ability to compute good approximations to MPE will depend on our ability to approximate, within the span of the chosen basis functions, the optimal value function when competitors use the candidate equilibrium strategy. In particular, as we improve the approximation architecture (for example, by adding more basis functions), the approximation to MPE should improve. We illustrate this with a few examples.

Basis functions. We describe a generic family of basis functions that we believe allows us to systematically construct increasingly sophisticated approximations to the optimal value function. We will show that this family will serve to approximate MPE accurately in important classes of EP-style models. We also discuss the relation between this family of basis functions and other commonly used sets of basis functions, such as those based on polynomials and state aggregation.

For a given set $C \subset \mathcal{X}$, let us denote by $s(C)$ the vector defined by the components of s in C. In particular, $s(C)$ yields the histogram of firms restricted to individual firm states in C. For example, if $\mathcal{X} = \{0, \dots, 3\}$, a given industry state $s = (6, 7, 9, 5)$, and $C = \{1, 2\}$, then $s(C) = (7, 9)$. Now for a given individual state x, let $\mathcal{C}_x$ be a set of subsets of $\mathcal{X}$. For instance, we may have $\mathcal{C}_x = \{\{i\} : i \in \mathcal{X}\}$, or we might have $\mathcal{C}_x = \{\{i, j\} : i, j \in \mathcal{X}\}$, or, for that matter, we may even consider $\mathcal{C}_x = \{\mathcal{X}\}$.

Associating every individual state $x \in \mathcal{X}$ with such a set $\mathcal{C}_x$, we consider approximations to the value function of the form

$$V^*_{\mu,\lambda}(x, s) \approx \sum_{C \in \mathcal{C}_x} f^C(x, s(C)),$$

where each f^C is an arbitrary function of its arguments. We next consider a series of examples that should make the flexibility of this architecture apparent.

(i) No approximation. Notice that if $\mathcal{C}_x = \{\mathcal{X}\}$ for all x, this is not an approximation at all; we may capture the entire value function exactly. In particular, we have

$$V^*_{\mu,\lambda}(x, s) \approx f^{\mathcal{X}}(x, s),$$

which is not an approximation. Of course, the corresponding set of basis functions will be far too large to be useful (i.e., it will require as many numbers to encode as the size of the state space $\mathcal{Y}$).

(ii) Separable approximations. Consider taking $\mathcal{C}_x = \{\{i\} : i \in \mathcal{X}\}$ for every x. This corresponds to an approximation of the form[9]

$$V^*_{\mu,\lambda}(x, s) \approx \sum_{j \in \mathcal{X}} f^{\{j\}}(x, s(j)).$$

For a given state (x, s), here we seek to approximate the value function by a sum of $|\mathcal{X}|$ functions, each of which is an arbitrary function of the firm's own state, x, and the number of firms at a specific individual state, $s(j)$. Note that the approximation is *separably* additive over functions that each depend on the number of firms at a particular individual firm state. In this approximation, there is one basis function per pair of individual states (x,j); each such

[9] In practice, our approximation also includes a constant term, independent of (x, s). The use of such a constant is motivated by the theoretical results of this article; see web Theorem C.1.

function is specified by $N + 1$ numbers. Hence, the overall approximation can be encoded by $|\mathcal{X}|^2 \times (N + 1)$ numbers, a dramatic reduction from the size of the state space. To make this more concrete, suppose $|\mathcal{X}| = 2$. Then, the separable approximation is of the form $V^*_{\mu,\lambda}(x, s(0), s(1)) \approx f^{\{0\}}(x, s(0)) + f^{\{1\}}(x, s(1))$, where $f^{\{0\}}$ and $f^{\{1\}}$ are arbitrary functions. The first term corresponds to a contribution to the approximate value function that depends on the firm's own state and the number of firms in individual state 0; the second one depends on the number of firms in individual state 1. Note that the fully flexible value function allows for any function of (x, s(0), s(1)), whereas this approximation only allows for the separably additive form.

(iii) Moment approximations. One can also consider moment-based approximations. Recall that the value function depends on the firm's individual state and the industry state; the latter can be viewed as a distribution of firms over individual states. Here we attempt to approximate the value function by a function of the firm's own state and a few (unnormalized) moments of this distribution. Specifically, a moment-based approximation takes the form

$$V^*_{\mu,\lambda}(x, s) \approx r_x + \sum_{k\in\mathcal{K}} r_{x,k} \left(\sum_{j\in\mathcal{X}} s(j) j^k \right),$$

where $\mathcal{K}$ is a set of (typically positive) integers. Notice that $\sum_{j\in\mathcal{X}} s(j)j^k$ is the kth unnormalized moment of the distribution over individual states that describes industry state s. We may interpret the above approximation as a linear combination of the moments in $\mathcal{K}$ where the weights of this linear combination are specific to the firm's own state x.

It is not difficult to see that this intuitive approximation is nothing but a special case of the separable approximation described above. In particular, one sees that the moment approximation can be recovered by defining the functions $f^{\{j\}}(x, s(j))$ in a separable approximation according to

$$f^{\{j\}}(x, s(j)) = \frac{r_x}{|\mathcal{X}|} + \sum_{k\in\mathcal{K}} r_{x,k} s(j) j^k.$$

Moment approximations are a particular case of separable approximations. With some abuse of terminology, in the rest of the article we refer to a separable approximation to the fully general separable one introduced in point (ii) above, and to a moment approximation to a specific moment-based one introduced in point (iii) above.

Evidently, by picking appropriate sets $\mathcal{C}_x$, the basis functions described can be used to capture a rich array of approximations. In particular, one could consider $\mathcal{C}_x = \{\{i, j\} : i, j \in \mathcal{X}\}$, for each x. This approximation architecture is more general than the separable one described above. Specifically, this corresponds to an approximation of the form

$$V^*_{\mu,\lambda}(x, s) \approx \sum_{i,j\in\mathcal{X}} f^{\{i,j\}}(x, s(i), s(j)).$$

Going further, one could consider $\mathcal{C}_x = \{\{i, j, k\} : i, j, k \in \mathcal{X}\}$ and so forth. Of course, richer specifications require a larger number of basis functions.

The family of basis functions we have described is easily expressed in the form described at the start of this section, where we approximate the value function $V^*_{\mu,\lambda}$ with Φr for a set of basis functions $\Phi_i : \mathcal{Y} \to \mathbb{R}$, $i = 1, 2, \ldots, K$. This requires introducing appropriately defined indicator functions.

As a concrete example, the separable approximation to the value function can be encoded as follows. For all $i, j \in \mathcal{X}$ and $k \in \{0, 1, \ldots, N\}$, define the indicator function

$$\Phi_{i,j,k}(x, s) = \begin{cases} 1 & \text{if} \quad x = i \quad \text{and} \quad s(j) = k \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all} \quad (x, s) \in \mathcal{Y}.$$

That is to say, $\Phi_{i,j,k}(\cdot, \cdot)$ is an indicator for the state where a firm is in individual state i and the industry state has k competitors at individual state j. Then, *any* function of the form $f^{\{j\}}(x, s(j))$ can be written as

$$f^{\{j\}}(x, s(j)) \triangleq \sum_{i\in\mathcal{X}, k\in\{0,\ldots,N\}} \Phi_{i,j,k}(x, s) r_{i,j,k},$$

with the appropriate weights $r_{\cdot,j,\cdot}$. It follows that any separable approximation may be succinctly expressed as Φr, where Φ is a matrix in $\{0, 1\}^{|\mathcal{Y}|\times|\mathcal{X}|^2\cdot(N+1)}$ with a column, $\Phi_{i,j,k} : \mathcal{Y} \to \{0, 1\}$ for each $i, j \in \mathcal{X}$ and $k \in \{0, 1, \ldots, N\}$, and $r \in \mathbb{R}^{|\mathcal{X}|^2\cdot(N+1)}$.

Selecting an appropriate set of basis functions is problem dependent. In what follows, we test the general separable and moment-based approximation architectures in the quality ladder and capacity models introduced in Section 3 and described in detail in Section 6. To do this, we first solve for the exact MPE for the case of $N = 3$. Then, we compute the approximation to MPE using the moment-based and separable approximating architectures described above, setting c in the mathematical program to be the vectors of ones. [10]

Quality ladder model. Figure 1 displays the value functions associated to each set of basis functions and to the exact MPE for the quality ladder and capacity competition models in the upper and lower panels of the figure, respectively. We observe that for the quality ladder model, the MPE value function has an intuitive pattern; for a fixed industry state it increases with the firm's own quality level, and for a fixed firm's own quality level it decreases with the "competitiveness" of the industry state. Note that even an approximation with just two moments is able to capture these patterns and is quite accurate. Of course, as we enrich the basis functions by adding more moments or by moving to the fully flexible separable specification, the approximation improves even more.

The upper panel in Figure 2 shows the investment strategies for a different set of basis functions and the exact MPE. Except for a boundary effect at the firm's smallest quality level (i.e., the smallest value taken on by x), the MPE investment strategy—to a first order— decreases with the firm's own quality level and the competitiveness of the industry state. Again, even the architecture with two moments captures this pattern and provides a reasonably accurate approximation.

The results suggest that for this model, a set of basis functions that only depends on a few features of the industry state is enough to obtain a good approximation. In fact, a linear regression of the single-period profit function and the MPE value function against a constant, the firm's own state, and the first moment of the industry state yields an R^2 of 0.9 and 0.97, respectively; the impact of the competitors' state in equilibrium outcomes can indeed be summarized by a few simple statistics.

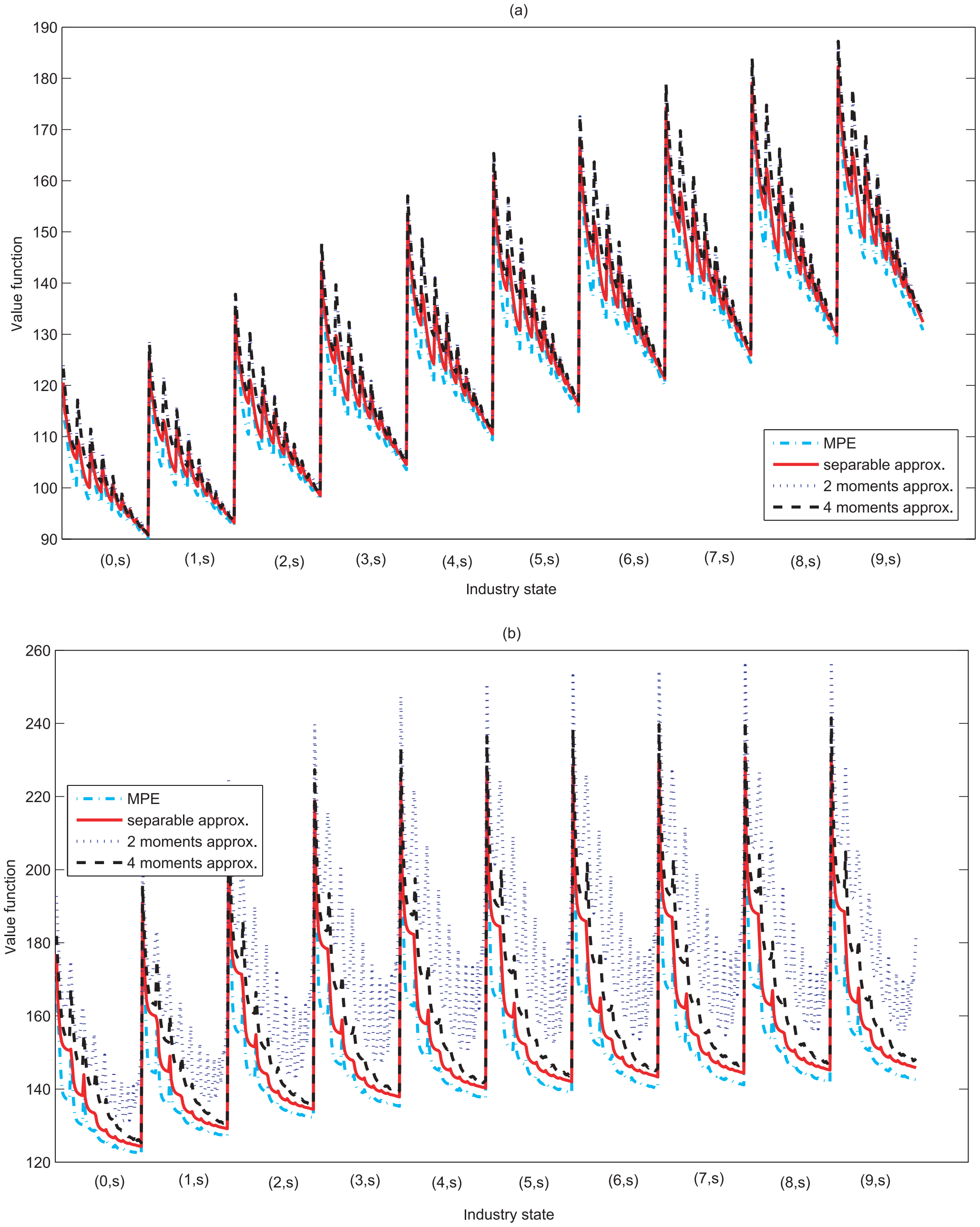
Capacity competition model. The MPE value function of this model exhibits rougher patterns (see lower panel of Figure 1). For a fixed industry state, the value function first increases as a function of the firm's own state and then after individual state 4 it basically becomes flat. For a fixed firm's own state, it decreases with, the competitiveness of the industry state, but after a point it also flattens. This is because the monopoly quantity for this model is between the capacities in individual states 3 and 4, so firms will never produce more than that. Hence, in terms of single-period profits, competitors beyond individual state 4 are all equivalent. This effect is also expressed in the value function. In principle, due to depreciation, firms in larger states are tougher dynamic competitors, because they are less likely to fall below the monopoly quantity capacity state in the short run. However, this effect is alleviated by the fact that firms invest even beyond the monopoly quantity state to try to prevent this from happening (see Figure 2, lower panel).

The previous discussion already suggests that the actual state of competitors has a much larger impact on equilibrium outcomes. For example, two competitors in state 4 are much tougher

[10] In these examples, the approximations based on the first two and four moments include 30 and 50 basis functions, respectively, whereas the separable approximation includes 400 basis functions. The size of the state space is 660.

FIGURE 1

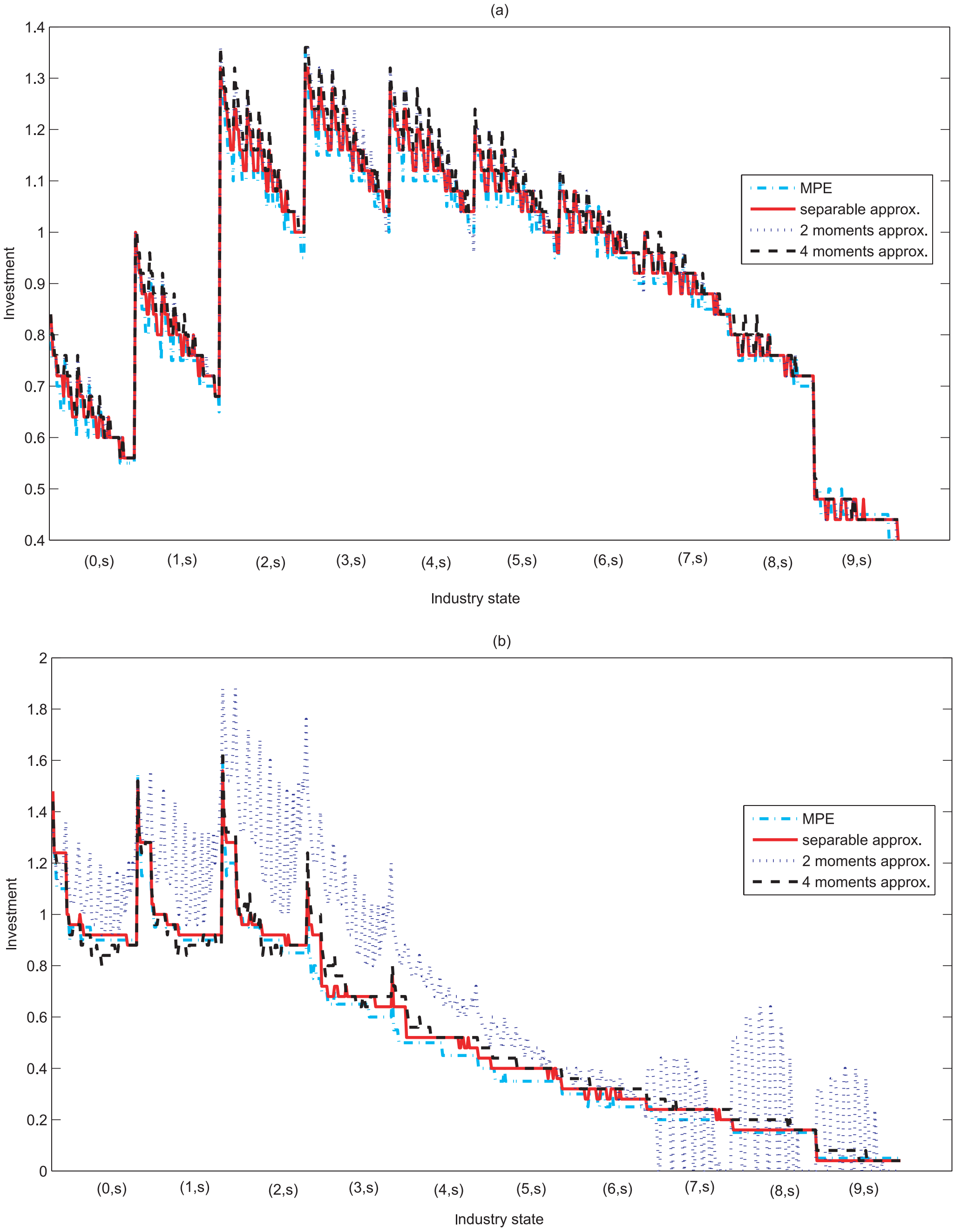
VALUE FUNCTION APPROXIMATION FOR DIFFERENT SETS OF BASIS FUNCTIONS AND COMPETITION MODELS



The upper graph compares approximations from separable and moment-based architectures for the quality ladder model. The lower graph compares approximations from separable and moment-based architectures for the capacity competition model. States are ordered as $\{(x = 0, s) : s \in \mathcal{S}\}$, $\{(x = 1, s) : s \in \mathcal{S}\}, \ldots$. The industry states in $\mathcal{S}$ are roughly listed in increasing order of "competitiveness", where a more competitive state means that it has more rivals and/or rivals in larger states.

FIGURE 2

INVESTMENT STRATEGY APPROXIMATION FOR DIFFERENT SETS OF BASIS FUNCTIONS AND COMPETITION MODELS



The upper graph compares strategies from separable and moment-based architectures for the quality ladder model. The lower graph compares strategies from separable and moment-based architectures for the capacity competition model. States are ordered as $\{(x = 0, s) : s \in \mathcal{S}\}$, $\{(x = 1, s) : s \in \mathcal{S}\}, \ldots$. The industry states in $\mathcal{S}$ are roughly listed in increasing order of competitiveness, where a more competitive state means that it has more rivals and/or rivals in larger states.

than one competitor in state 8, even though their first unnormalized moments are the same. In fact, as previously discussed, in terms of static competition, one competitor in state 8 is equivalent to it being in state 4. It is perhaps not surprising then that the R^2 of the two regressions mentioned above are much lower: 0.4 and 0.6, respectively.

Our discussion and the results in Figures 1 and 2, lower panels, suggest that for this model, approximations based on a few moments do not work as well; a more detailed representation is needed to obtain accurate approximations. For this model, we apparently need the fully separable approximation architecture to get reasonable accurate approximation to the MPE value function and strategy function.

Discussion. The simple exercise described above suggests that the selection of good basis functions is an important contributor to the success of our approach. The quality of a given approximating architecture depends on the specific model it intends to approximate. Choosing a good set of basis functions requires understanding what features of the state may have the largest impact on the value function and optimal strategy, and a fair amount of trial and error. The comparisons and linear regressions described in the previous sections together with experiments like the ones presented in Section 6 can support this process in practice. In Section 6, we will show that the separable approximation architecture discussed above is effective for the class of EP-style models we study; that architecture seems to capture MPE strategic interactions well. As described above, there is a natural extension to this set of basis functions that may be used if a richer architecture is called for.

It is worth digressing to discuss other approximation architectures suggested in the economics literature. Judd (1998) proposes the use of polynomials to approximate the value function in low-dimensional dynamic programming problems with continuous state spaces. This approach can be useful in a setting with a relatively small number of firms but where individual states are continuous. The present article focuses on a complementary setting with a large number of firms but a finite state space. That said, the architectures proposed by Judd can also be used to good effect in our framework. For example, in the separable approximation, there are no restrictions imposed over the functions $f^{\{j\}}(x, s(j))$; in particular, they could be polynomials. Of course, each of these functions only depends on the number of firms in a particular individual state, $s(j)$. For a larger class of polynomials, one could consider, for example, $\mathcal{C}_x = \{\{i, j\} : i, j \in \mathcal{X}\}$, in which one can include polynomial functions that depend on $s(i)$ and $s(j)$, for $i, j \in \mathcal{X}$.

Another common approximation architecture is based on state aggregation (Tsitsiklis and Van Roy, 1996). Here, the state space is partitioned into sets, and all states in a specific set share the same value for the value function. This architecture can be easily encoded using indicator basis functions. However, we believe that the set of basis functions we introduced in this section provides more flexibility and is more appropriate to approximate the value function in the class of models we study.

Moment-based approximations have been previously used in large-scale stochastic control problems that arise in macroeconomics (Krusell and Smith, 1998). Distinct to the present work, the structure of the specific problems there permits an approximation not only of the value function but also of the *dynamics* of agents. This effectively reduces the original dynamic programming problems to "aggregate" dynamic programs in a reduced state space for which a tailor-made algorithm is developed for equilibrium computation.

□ **Weight vector *c* and constraint sampling.** In this section, we first describe the importance of the state relevance weight vector c. Then, we describe a constraint sampling scheme. We also provide a numerical example to illustrate how these steps work in practice.

Approximation error: state relevance weight vector c. If the state space is large, it is unlikely that a parsimonious set of basis functions allows approximating the value function uniformly well over the entire state space. This is shown, for example, in Figures 1 and 2, where the approximation errors to the value function are larger in some portions of the state space. This is likely to be exacerbated in larger state spaces. Therefore, it is useful to point out that theory suggests that the state relevance weight vector c trades off approximation error across different states; the mathematical program will provide better approximations to the value function for states that have larger weights in the objective function. By choosing different c vectors, the user can effectively reduce approximation error in different parts of the state space and hopefully obtain accurate approximation for the set of "relevant" states. We formalize this notion in web Appendix C.2.

For example, suppose we are interested in the short-run dynamic behavior of an industry starting from a given initial state. More specifically, we want to assess how an industry would evolve over a few years after a policy or environmental change such as a merger. Then, relevant states are ones that are visited with high probability starting from the given initial condition over a horizon of say T periods. In this case, the vector c should assign larger weights to this set of states.

As another example, suppose we are interested in the long-run behavior of the industry (that is independent of the initial condition). Then, relevant states are ones that have significant probability of occurrence under the invariant distribution of the Markov process that describes the industry evolution. In this case, c should be the invariant distribution.

In practice, the weight vector c will be required in computing an approximate best response to some current strategy in the course of the use of an iterative best response scheme for equilibrium computation such as Algorithm 1. In that case, the distributions alluded to above may be selected as those corresponding to the incumbent strategy in the algorithm, and can be computed with forward simulation. An important observation is that because these strategies change in the course of the algorithm, the set of relevant states and, hence, the weight vector c, should also change as the algorithm progresses.

Reducing the number of constraints. In the previous sections, we discussed how to reduce the number of variables using value function approximation and how to trade off approximation error across different states. However, the number of constraints is still prohibitive. In this section, we describe how a constraint sampling scheme alleviates this difficulty.

Value function approximation reduced the number of variables of the program (3). To deal with the large number of constraints, we will simply sample states from $\mathcal{Y}$ and only enforce constraints corresponding to the sampled states. Now, because the number of variables common to all constraints in (4) is small, that is, it is simply the number of basis functions, K, we will see that the resulting "sampled" program will have a tractable number of variables and constraints.

Given an arbitrary sampling distribution over states in $\mathcal{Y}$, ψ, one may consider sampling a set $\mathcal{R}$ of L states in $\mathcal{Y}$ according to ψ. We consider solving the following relaxation of (4):

$$\begin{array}{ll} \min & c'\Phi r \\ \text{s.t.} & (T_{\mu,\lambda}\Phi r)(x,s) \le (\Phi r)(x,s), \quad \forall (x,s) \in \mathcal{R}. \end{array} \tag{5}$$

We note that the program (5) is a *convex* optimization problem: it has a convex objective and convex constraints. As such, provided L is not too large, one can, in principle, solve this program directly (see Boyd and Vandenberghe, 2004). Intuitively, a sufficiently large number of samples L should suffice to guarantee that a solution to (5) satisfies all except a small set of constraints in (4) with high probability. In fact, theory suggests that if the distribution chosen mimics the choice of c suggested in the previous section (so it focuses on the set of relevant states), L can be chosen independently of the total number of constraints in order to achieve a desired level of performance. By sampling a sufficiently large, but tractable, number of constraints via an appropriate sampling distribution, one can compute an approximate best response via (5) whose quality is similar to that of an approximate best response computed via the intractable program (4). We illustrate this point with an example and present theoretical support in web Appendix C.3.

TABLE 1 Weighted Relative Errors of Value Function for the Capacity Competition Model, Using Moment-Based Approximation with Two moments When $N = 4$ and $c = c_{LR}$

	$r = 500$	$r = 250$	$r = 100$
ϵ_r^V	0.0180	0.0121	0.1145

We point out that different selections of the weight vector c and the constraint sampling distribution can result in different approximations to MPE. In this section and the previous one, we have suggested appropriate selections for these quantities that are supported by theory. Moreover, computational experimentation with the models presented in Section 3 confirmed that, compared to other alternatives, the selections we suggest consistently provide the best approximations to the MPE computed with our best response algorithm.

Examples. Consider the capacity competition model introduced in Section 3 and described in detail in Section 6, with $N = 4$. In this setting, one has that $|\mathcal{Y}| = 2860$. We solve for the MPE and use it to estimate the long-run distribution of industry states over $\mathcal{Y}$ (through simulation). Let c_{LR} denote such an estimate. Let $\mathcal{Y}_{LR}(r) \subseteq \mathcal{Y}$ include the r most frequently visited states according to c_{LR}.

We solve (5) for several values of $|\mathcal{R}|$, where we consider (μ, λ) to be the MPE, and $c = c_{LR}$. That is, we compute an approximate best response to the MPE using the mathematical program (5) and taking $c = c_{LR}$. To illustrate the concepts discussed above, we use the moment-based approximating architecture with $|\mathcal{K}| = 2$ (so we consider the first two unnormalized moments).

Let V^r denote the resulting best response approximation value function when the sampled constraints in formulation (5) are $\mathcal{R} = \mathcal{Y}_{LR}(r)$ for $r = 500, 250, 100, |\mathcal{Y}|$. With these, we compute the following weighted approximation relative errors:

$$\epsilon_r^V := \sum_{(x,s)\in\mathcal{Y}} c_{LR}(x,s) \cdot |V^{|\mathcal{Y}|}(x,s) - V^r(x,s)|/V^{|\mathcal{Y}|}(x,s), \quad r \in \{500, 250, 100\}.$$

The errors above quantify the quality of the approximation produced when solving the approximate mathematical program using only the constraints considered in $\mathcal{R}$ as opposed to all constraints.

Table 1 illustrates the results of our numerical experiments. There, one appreciates that reducing the number of constraints in formulation (5) does impact the quality of the approximation, although the impact is moderate when weighting according to the long-run distribution induced by MPE. Even with a tenth of the constraints, the approximation does not degrade much.[11]

□ **Discretization and a tractable linear program.** The mathematical program (5) has *non linear* constraints. Although, in principle, one could attempt to solve that convex non linear program directly, in this section we propose a finite linear programming approximation to (5) that is much simpler to solve. Our approximation will require discretizing the allowable investment levels. In addition, to obtain a linear formulation, we will approximate the continuous sell-off value random variable by a discrete random variable. We note, however, that in web Appendix B, we introduce a heuristic to solve our linear program that *does not require* the discrete approximation to the sell-off value; it only requires discretizing the allowable investment levels. For completeness, we present both discretizations in this section.

Discretizing sell-off Values. We replace the continuous valued sell-off value random variable κ by a discrete-valued random variable $\hat{\kappa}$ defined as follows: $\hat{\kappa}$ takes values uniformly at random

[11] Of course, generally we will not have access to the MPE strategy. Instead, as previously mentioned, in practice we sample states using the incumbent strategy in the algorithm.

in the set $\hat{\mathcal{K}} = \{k_1, k_2, \dots, k_n\}$, where k_j is the largest quantity satisfying $\mathcal{P}[\kappa < k_j] \le \frac{n+1-j}{n}$ for $j = 1, \dots, n$. Here n is a parameter that will control the fineness of the discretization.

Discretizing investment levels. As opposed to allowing investments in $[0, \bar{\iota}]$, we only allow for investments in the set

$$\mathcal{I}^{\epsilon} = \{0, \epsilon, 2\epsilon, \dots, \lfloor (\bar{\iota})/\epsilon \rfloor \epsilon\},$$

where $\epsilon > 0$ is a parameter that controls the fineness of our discretization.

The above discretizations are tantamount to the approximation

$$(T_{\mu,\lambda}\Phi r)(x,s) \sim \pi(x,s) + \frac{1}{n}\sum_{i=1}^{n} \big(k_i \vee \big(C^{\epsilon}_{\mu,\lambda}\Phi r\big)(x,s)\big) \triangleq \big(T^{\epsilon,n}_{\mu,\lambda}\Phi r\big)(x,s),$$

where $(C^{\epsilon}_{\mu,\lambda}\Phi r)(x,s) = \max_{\mu'(x,s):\iota'(x,s)\in\mathcal{I}^{\epsilon}} (C^{\mu'}_{\mu,\lambda}\Phi r)(x,s)$. We then consider solving the following program instead of (5):

$$\begin{aligned} \min \quad & c'\Phi r \\ \text{s.t.} \quad & \big(T^{\epsilon,n}_{\mu,\lambda}\Phi r\big)(x,s) \le (\Phi r)(x,s), \quad \forall (x,s) \in \mathcal{R}. \end{aligned} \tag{6}$$

Two questions arise regarding the program (6). First, what impact does discretization have on our accuracy in solving the best response problem? Second, why is (6) any easier to solve? The first question is answered in web Appendix A, where we show that the approximation does not degrade much with a fine-enough discretization of the sell-off value distribution and investment levels.

We note that in an exact MPE computation, introducing discrete investment levels may destroy the existence of pure-strategy equilibrium as shown by Doraszelski and Satterthwaite (2010). In this case, if mixed strategies are not allowed when computing exact MPE, one may need to relax the stopping criteria in Algorithm 1, as we now explain. It is reasonable to expect that with discrete investment levels, Algorithm 1 may jump between adjacent investment levels in consecutive iterates, such that $\|\iota_i - \iota_{i+1}\|_\infty = \epsilon$ (ϵ is the parameter that controls for the fineness of the discretization in investment). If this happens, one would stop and consider $(\iota_i + \iota_{i+1})/2$ as an approximation to a pure-strategy MPE. In this case, one may expect that as ϵ becomes small, one gets closer to a pure-strategy MPE of the continuous model.

Now, in our approach, we view the discretization of investment levels as one additional layer of approximation to approximate MPE of the original continuous model. We note that with the stopping criteria that we use (see web Appendix B for details), our approximation scheme always converged to a pure investment strategy. Moreover, as we show in the numerical experiments presented in Section 6, our approach with value function approximation, constraint sampling, and discrete investment levels produces accurate approximations to MPE of the original continuous model.

As for the second question of what the advantages are of solving (6), we can show that in fact (6) is equivalent to a linear program, which is substantially simpler to solve than the non linear program (5). We describe this equivalent linear program next.

Notice that by introducing auxiliary variables $u(x,s) \in \mathbb{R}^n$, the constraint $(T^{\epsilon,n}_{\mu,\lambda}V)(x,s) \le V(x,s)$ is equivalent to the following set of constraints[12]:

$$\begin{aligned} & \pi(x,s) + \frac{1}{n}\sum_{i=1}^{n} u(x,s)_i \le V(x,s) \\ & \max_{\mu'(x,s):\iota'(x,s)\in\mathcal{I}^{\epsilon}} C^{\mu'}_{\mu,\lambda}V(x,s) \le u(x,s)_i \qquad \forall i \in \{1,\dots,n\} \\ & k_i \le u(x,s)_i \qquad\qquad\qquad\qquad\qquad\quad \forall i \in \{1,\dots,n\}. \end{aligned} \tag{7}$$

[12] By equivalent, we mean that the set of values of $V(x, s)$ that satisfy the constraint is identical to the set of values of $V(x, s)$ that satisfy (7).

These constraints, except for the second one, are linear in the set of variables u and V. However, it is easy to see that, for each i, the second non linear constraint above is equivalent to a set of $|\mathcal{I}^{\epsilon}|$ linear constraints:

$$-d\iota' + \beta E_{\mu,\lambda}[V(x_1, s_1)|x_0 = x, s_0 = s, \iota_0 = \iota'] \leq u(x, s)_i, \quad \forall \iota' \in \mathcal{I}^{\epsilon}.$$[13]

Hence, (6) is, in fact, equivalent to the following linear program:

$$\begin{aligned}
\min \quad & c'\Phi r \\
\text{s.t.} \quad & \pi(x, s) + \frac{1}{n}\sum_{i=1}^{n} u(x, s)_i \leq \Phi r(x, s) && \forall\, (x, s) \in \mathcal{R} \\
& -d\iota' + \beta E_{\mu,\lambda}[\Phi r(x_1, s_1)|x_0 = x, s_0 = s, \iota_0 = \iota'] \leq u(x, s)_i && \forall \iota' \in \mathcal{I}^{\epsilon},\ \ \forall i \in \{1, \ldots, n\}, \forall (x, s) \in \mathcal{R} \\
& k_i \leq u(x, s)_i && \forall i \in \{1, \ldots, n\},\ \ \forall (x, s) \in \mathcal{R}.
\end{aligned} \tag{8}$$

In summary, we have constructed a linear program with a tractable number of variables and constraints to approximate the best response value function. One last potential difficulty to solving this program is the expectation over next-period states that we need to compute in the left-hand side of the constraints in (6); as pointed out by Doraszelski and Judd (2010), this involves a high-dimensional sum. We show in web Appendix B.2, however, that this sum gets significantly simplified with the basis functions we use.

□ **The overall algorithm for approximate MPE computation.** This section will integrate the approximate dynamic programming approach we have developed for the computation of an approximate best response in the previous section into Algorithm 1. In this way, we build a tractable algorithm to approximate MPE. A more detailed version of the overall algorithm is presented in web Appendix B; this will likely be of interest to a reader implementing the approach.

We briefly recall the key inputs to the algorithm. In addition to the specification of the industry model itself, the main inputs to the algorithm are the following.

- An *initial investment/exit strategy* denoted (μ^c, λ^c). There are a number of choices one may make here based on an understanding of the industry at hand. A generic choice might be a "myopic" strategy, where both entrants and incumbents consider only the present period in making their decisions.
- An *approximation architecture* to approximate the value function of an incumbent. Recall that we encode this as basis functions $\Phi_i : \mathcal{Y} \to \mathbb{R}$, $i = 1, 2, \ldots, K$. Again, problem-specific knowledge can guide the selection of a suitable architecture, and previously in the article we introduced what we consider a generically useful approach to producing a "good" approximation architecture.
- Parameters related to algorithm performance and precision; these include, for instance, a tolerance ϵ for our stopping criterion, and parameters specifying the extent and nature of sampling used at various points in the algorithms, among other parameters.

We now describe the overall algorithm itself. We present the various "logical" phases of the algorithm here; the interested reader can find pseudo code for the algorithm in web Appendix B.

(i) Start with an initial entry/investment strategy $(\mu, \lambda) = (\mu^c, \lambda^c)$ as discussed above.
(ii) Sample states. Sample industry states assuming all firms use this strategy. Call the sampled set of industry states $\mathcal{R}$, the set of "relevant states." Moreover, let c be the empirical distribution over states induced by the samples collected. For example, if one is interested in the long-run behavior of the industry, an appropriate sampling procedure will produce samples that

[13] Note that for a fixed action ι' the expectation operator is linear in the set of variables V.

can safely be assumed to arise from the steady-state distribution induced by the incumbent strategy (μ, λ).

(iii) Solve approximate dynamic programming math program. As discussed in our development of the approximate dynamic programming approach, we next compute an approximate best response by solving the mathematical program (5):

$$\begin{aligned} \min \quad & c'\Phi r \\ \text{s.t.} \quad & (T_{\mu,\lambda}\Phi r)(x, s) \leq (\Phi r)(x, s), \quad \forall (x, s) \in \mathcal{R}. \end{aligned}$$

Recall that the above program is a finite-dimensional convex optimization problem; nonetheless, we might find it advantageous to employ a simpler heuristic solution procedure in solving it. One such procedure, based on discretization that yields a linear program, was described in Section 5. Yet another procedure that does not require that we discretize the sell-off value is described in web Appendix B. There, we only discretize investment levels and propose solving a sequence of linear programs to find an approximate best response.

(iv) Compute implied approximate best response. Using the solution to the above program, compute the implied best response strategy $(\hat{\mu}, \hat{\lambda})$. Note that this and for other steps such as building the approximate math program, we need access to the current incumbent strategy in the algorithm. However, due to the curse of dimensionality, storing such a strategy is generally prohibitive. Instead, we show that by using all past solutions (r) to the math program above in the course of the algorithm, we can in fact compute and recover the implied best response strategy; this does not add significant complexity. An algorithm to compute this implied best response is described in detail in web Appendix B.

(v) Decide whether to stop or iterate. Decide whether to stop based on whether the earnings of an incumbent who chooses to unilaterally deviate to the computed best response are small in an appropriate, weighted sense. In particular, evaluate

$$\Delta \triangleq c^{\top} \left| V^{\hat{\mu}}_{\mu,\lambda} - V_{\mu,\lambda} \right|$$

and stop if Δ is sufficiently small, say smaller than ϵ. In the event that $\Delta > \epsilon$, use $\hat{\mu}, \hat{\lambda}$ as the incumbent strategy and go back to the second step above. The stopping criteria are motivated by our theoretical results.

□ **Theoretical guarantees.** We conclude this section by briefly describing the theoretical guarantees we can offer for our approach. Web Appendices C and D present this theory in detail.

We first develop an extension of the theory developed in de Farias and Van Roy (2003, 2004) that lets us bound the magnitude of the increase in a firm's expected discounted profits if it unilaterally deviated to an optimal strategy from that produced by the approximate dynamic programming approach. In particular, these guarantees allow us to provide a stopping criterion under which our algorithm would terminate at what is essentially an ϵ equilibrium (Fudenberg and Tirole, 1991). Our theory demonstrates that the "ϵ" here crucially depends on the expressivity of the approximation architecture, among other algorithmic parameters. More specifically, the ϵ is guaranteed to be small if a good approximation to the value function corresponding to our candidate equilibrium strategy is within the span of our chosen basis functions.

We then demonstrate a relationship between our notion of ϵ equilibrium and approximating equilibrium strategies that provides a more direct test of the accuracy of our approximation. In web Theorem D.1 we show that as we improve our approximation so that a unilateral deviation becomes less profitable (e.g., by adding more basis functions), we indeed approach an MPE. The result is valid for general approximation techniques, and we anticipate it can be useful to justify other approximation schemes for dynamic oligopoly models or even in other contexts.

TABLE 2 **Default Parameters for Numerical Experiments**

Parameter	β	δ	γ	b	$\tilde{\kappa}$	$\tilde{\phi}$	$\overline{x}$	x^e	d
Value	0.925	0.70	0.00	3.00	30.00	300.00	9	1	1.00

6. Computational experiments

■ In this section, we conduct computational experiments to evaluate the performance of our algorithm in situations where either we can compute an MPE or a good approximation is available. We use both the quality ladder and the capacity competition models described in Section 3.[14] We start by describing our models in detail.

□ **Details of specific models.** In this section, we provide details regarding the models described in Section 3.

Sell-off and entry cost distributions. We consider exponentially distributed random variables to model both the sell-off value and the entry cost. In particular, in each time period, each potential entrant *i* will observe a random positive entry cost ϕ_{it} exponentially distributed with mean $\tilde{\phi}$. Also, each period, each incumbent firm i observes a positive random sell-off value κ_{it} exponentially distributed with mean $\tilde{\kappa}$.

Transition dynamics. Following Pakes and McGuire (1994), a firm that invests a quantity ι is successful with probability $(\frac{b\iota}{1+b\iota})$, in which case the quality of its product increases by one level. The firm's quality level depreciates one state with probability δ, independently each period. Independent of everything else, every firm has a probability γ of increasing its quality by one level. Hence, a firm can increase its quality even in the absence of investment.[15] If the appreciation shock is unsuccessful, then the transitions are determined by the investment and depreciation processes. Combining the investment, depreciation, and appreciation processes, it follows that the transition probabilities for a firm in state x that invests ι are given by

$$\mathcal{P}[x_{i,t+1} = y \mid x_{it} = x, \iota] = \begin{cases} (1-\gamma)\dfrac{(1-\delta)b\iota}{1+b\iota} + \gamma & \text{if } \; y = x+1 \\ (1-\gamma)\dfrac{(1-\delta)+\delta b\iota}{1+b\iota} & \text{if } \; y = x \\ (1-\gamma)\dfrac{\delta}{1+b\iota} & \text{if } \; y = x-1\,. \end{cases}$$

Parameter specification. In practice, parameters would either be estimated using data from a particular industry or chosen to reflect an industry under study. We use a set of representative parameter values, summarized in Table 2. We keep these parameters fixed for all experiments, unless otherwise stated. The values of δ and b are set as in Pakes and McGuire (1994). We also set the mean entry cost to be much higher than the mean sell-off value.

Note that in most applications the profit function would not be specified directly but would instead result from a deeper set of primitives that specify a demand function, a cost function, and a static equilibrium concept. Next, we specify two models that we will use in our computational experiments. The entry, exit, and investment processes are kept the same for both of these models.

[14] Our implementation of the algorithm described in the previous section together with detailed documentation can be found on the authors' webpages.

[15] In our experiments, we eventually consider both settings where $\gamma = 0$ and $\gamma > 0$. We discuss this in more detail in Web Appendix B.1.

TABLE 3 Default Parameters for Quality Ladder Model

Parameter	m	c	Z	θ_1	θ_2	Y
Value	100.00	0.50	1.00	0.50	0.50	1.00

Profit function: quality ladder model. We consider an industry with differentiated products, where each firm's state variable represents the quality of its product. That is, x_{it} represents the quality of the product offered by firm i in period t. There are m consumers in the market. In period t, consumer j receives utility u_{ijt} from consuming the good produced by firm i given by

$$u_{ijt} = \theta_1 \ln\left(\frac{x_{it}}{Z} + 1\right) + \theta_2 \ln(Y - p_{it}) + \epsilon_{ijt}, \quad \forall i,\ j = 1, \dots, m,$$

where Y is the consumer's income, p_{it} is the price of the good produced by firm i at time t, and Z is a scaling factor. ϵ_{ijt} are i.i.d. Gumbel random variables that represent unobserved characteristics for each consumer-good pair. There is also an outside good that provides consumers an average utility of zero. We assume consumers buy at most one product each period and that they choose the product that maximizes utility. Under these assumptions, the demand system is a classic logit model. Considering a constant marginal cost of production c, there exists a unique Nash equilibrium in pure strategies, which can be computed by solving the first-order conditions of the pricing game (see Caplin and Nalebuff, 1991). Let $p^*(x)$ denote the price charge by a firm in state x in such an equilibrium[16]; expected profits are given by

$$\pi_m(x_{it}, s_t) = m\sigma(x_{it}, s_t, p^*)(p^*(x_{it}) - c)\ ,\ \forall i\ ,$$

where $\sigma(x_{it}, s_t, p^*)$ is the logit market share. That is,

$$\sigma(x_{it}, s_t, p^*) = \frac{(x_{it}/Z + 1)^{\theta_1}(Y - p^*(x_{it}))^{\theta_2}}{1 + \sum_{x=0}^{\overline{x}} s_t(x)(x/Z + 1)^{\theta_1}(Y - p^*(x))^{\theta_2}}.$$

We use a particular set of representative parameter values, summarized in Table 3, that we keep fixed for all experiments, unless otherwise stated.

Profit function: capacity competition model. This model is based on the quantity competition version of Besanko and Doraszelski (2004). We consider an industry with homogeneous products, where each firm's state variable determines its production capacity so that a firm in state 0 has a capacity of $\overline{q}(0) = q_{min}$, and a firm in state $\overline{x}$ has a capacity of $\overline{q}(\overline{x}) = q_{max}$. Capacity grows linearly between states 0 and $\overline{x}$. Investment increases this capacity. At each period, firms compete in a capacity-constrained quantity-setting game. There is a linear demand function $Q(p) = m(e - fp)$ and an inverse demand function $P(Q) = e/f - Q/(mf)$, where P represents the common price charged by all firms in the industry, Q represents the total industry output, and e, m, and f are positive and finite parameters. To simplify the analysis, we assume the marginal costs of all firms are equal to zero. Given the total quantity produced by its competitors $Q_{-i,t}$, the profit-maximization problem for firm i at time t is given by

$$\max_{0 \le q_{it} \le \overline{q}(x_{it})} P(q + Q_{-i,t})q_{it},$$

where $\overline{q}(x_{it})$ is the production capacity at individual state x_{it}. It is possible to show that a simple iterative algorithm yields the unique Nash equilibrium of this game $q_t^* = (q_t(0), \dots, q_t^*(\overline{x}))$,

[16] Our implementation of this model computes p^* by solving (using the Newton-Raphson method) the system of equations resulting from the first-order conditions to each firm's profit maximization.

TABLE 4 Default Parameters for Capacity Model

Parameter	m	q_{min}	q_{max}	e	f
Value	40.00	5.00	40.00	1.00	1/4

which is characterized by the following set of equations:

$$q^*(x_{it}) = \max\left\{0, \min\left\{\overline{q}(x_{it}), \frac{1}{2}\left(me + q^*(x_{it}) - \sum_{y\in\mathcal{X}} s_t(y)q^*(y)\right)\right\}\right\}, \quad \forall i \in s_t.$$

Profits for firm i are then given by

$$\pi_m(x_{it}, s_t) = P\left(\sum_{x\in\mathcal{X}} s_t(x)q^*(x)\right) q^*(x_{it}).$$

We use a particular set of representative parameter values, summarized in Table 4, that we keep fixed for all experiments, unless otherwise stated.

□ **Approximation architecture.** A key issue in our approach is the selection of an approximation architecture, that is, a set of basis functions. For the class of EP models we study, we propose using the *separable* approximation presented in Section 5. In such an architecture, one has a collection of $|\mathcal{X}|^2 \cdot (N+1)$ basis functions. This number will typically be substantially smaller than $|\mathcal{Y}|$, and hence the use of this approximation architecture makes, for instance, the linear program that we proposed solving at each step of our approximate best response algorithm a tractable program. For example, in models in which the state space has on the order of 10^{15} states, we only require thousands of basis functions.

Whereas the separable architecture allows for *general* functions of the number of firms in a specific inidividual state, in our numerical experiments, we also used a coarser architecture where these functions were restricted to be piecewise linear. Specifically, for instances with $N \geq 20$, we introduce this architecture by modifying our proposed linear program as follows. For a set $\mathcal{H} \subseteq \{0, \ldots, N\}$, define $l(j) = \max\{i \in \mathcal{H} : i \leq j\}$ and $u(j) = \min\{i \in \mathcal{H} : i \geq j\}$. We impose the following set of additional constraints:

$$r_{i,j,h} = \frac{u(h) - h}{u(h) - l(h)} r_{i,j,l(h)} + \frac{h - l(h)}{u(h) - l(h)} r_{i,j,u(h)} \quad \text{for all} \quad i \in \mathcal{X}, j \in \mathcal{X}, \quad \text{and} \quad h \notin \mathcal{H}.$$

That is, for each $i \in \mathcal{X}$, $j \in \mathcal{X}$, and $h \notin \mathcal{H}$, the variables $r_{i,j,h}$ (as defined in Section 5) are determined by linear interpolation. This procedure reduced the number of basis functions, and our numerical experience suggested it did not significantly degrade the accuracy of the approximation.[17]

□ **Comparing economic indicators of interest.** For the models introduced above, we show that our approximate linear programming (ALP) based algorithm with the proposed architecture provides accurate approximations to MPE behavior. For this purpose, we compare the outcome of our ALP approach to the outcome of computable benchmarks. Specifically, we first compare the strategy derived from our algorithm against MPE for instances with relatively small state spaces in which MPE can be computed exactly. Second, we compare the strategy derived from our algorithm against oblivious equilibrium, introduced by Weintraub, Benkard and Van Roy

[17] In our experiments with $|\mathcal{X}| = 10$, such a procedure resulted in a reduced approximating architecture with approximately 1200 basis functions for instances with more than 20 firms. Note that, in the absence of such constraints, the separable architecture would include approximately 2100 and 4100 basis functions for instances with 20 and 40 firms, respectively.

TABLE 5 Parameter Selection for Comparison with MPE

	Capacity Competition	
Parameters	High Inv.	Low Inv.
q_{min}	1	5
f	0.5	0.25
d	0.75	2.0
$\tilde{\phi}$	150	250
$\tilde{\kappa}$	50	75
	Quality Ladder	
	High Inv.	Low Inv.
θ_1	0.75	0.5
d	0.4	1.0
c	0.55	0.5
$\tilde{\phi}$	250	150
$\tilde{\kappa}$	100	80
γ	0.1	0.1

(2008), for instances with large numbers of firms and parameter regimes where OE is known to provide a good approximation.

Instead of comparing ALP strategies to our benchmark strategies directly, we instead compare economic indicators induced by these strategies. These indicators are long-run averages of various functions of the industry state under the strategy in question. The indicators we examine are those that are typically of interest in applied work; we consider average investment, average producer surplus, average consumer surplus, and average share of the *i*th largest firms (*Ci*), where the values for *i* to be examined will depend on the specific value of N (for example, if $N = 4$, one may be interested in examining $C2$, whereas if $N = 40$, one may also be interested in $C6$).[18]

Comparison with MPE. Exact computation of MPE is only possible when the state space is not too large. Therefore, we will begin by considering settings where the number of firms and the number of individual states are relatively small. For these instances, we will compare the strategy generated by our ALP-based algorithm with an MPE strategy. We compute MPE with Algorithm 1.

We consider several parameter regimes. First, we consider two regimes in the capacity competition model: one in which incentives to invest are strong, yielding a rich investment process, and one in which incentives to invest are weaker, yielding lower levels of investment. We also consider similar regimes for the quality ladder model. Table 5 depicts parameter selection for each instance. Web Appendix B specifies the algorithmic parameters used.

Table 6 reports the results for different values of N for which exact computation of MPE is feasible. There, we report MPE and ALP-based long-run statistics, and the percentage difference between them. Our ALP-based algorithm has a running time that is on the order of minutes. Exact computation of MPE took from a couple of seconds, for $N = 3$, to several hours, for $N = 5$.[19]

Our approximation provides accurate approximations of the economic indicators of interest in all instances. In fact, ALP-based indicators are always within 9.0% of MPE indicators, and

[18] Because the outcome of our algorithm is random, due to the sampling of constraints, ALP-based quantities reported in this subsection represent the average of five runs. In each run, industry evolution is simulated during 10^4 periods. The resulting sample of five data points is such that for each indicator, the ratio between the sample standard deviation and the sample mean is on average less than 1% and always less than 7%.

[19] All runs were performed on a shared cluster of 17 computers at Columbia Graduate School of Business. Each node has a 2.4 Ghz Intel (R) Xeon (R) CPU and 32 Gb of Ram memory. Our Java implementation used CPLEX 12.1.0 (facilitated by the IBM ILOG Academic Initiative) as a subroutine to solve the linear programs in the algorithms.

TABLE 6 Comparison of MPE and ALP-Based Indicators

				Long-Run Statistics					
Instance		Number of firms		Total Inv.	Prod. Surp.	Cons. Surp.	$C1$	$C2$	Entry Rate
Capacity competition model	High investment	$N = 3$	MPE	3.0879	17.6150	14.6262	0.5334	0.8531	0.2084
			ALP-Based	3.0587	17.8515	13.4731	0.5495	0.8640	0.2143
			% Diff.	0.95	1.34	7.88	3.03	1.28	2.86
		$N = 4$	MPE	3.3922	16.6884	17.4042	0.4313	0.7326	0.3250
			ALP-Based	3.2436	17.0408	16.3356	0.4462	0.7459	0.3349
			% Diff.	4.38	2.11	6.14	3.46	1.81	3.04
		$N = 5$	MPE	3.5304	15.6986	19.6373	0.3638	0.6385	0.4556
			ALP-Based	3.3697	16.1112	18.6626	0.3756	0.6513	0.4633
			% Diff.	4.55	2.63	4.96	3.25	2.01	1.71
	Low investment	$N = 3$	MPE	1.6292	34.4300	34.0271	0.4610	0.8017	0.1752
			ALP-Based	1.6000	34.7816	33.0713	0.4692	0.8082	0.1760
			% Diff.	1.79	1.02	2.81	1.78	0.82	0.47
		$N = 4$	MPE	1.4311	31.4584	40.0369	0.3625	0.6641	0.2934
			ALP-Based	1.4783	31.5736	39.7765	0.3651	0.6664	0.2965
			% Diff.	3.30	0.37	0.65	0.70	0.35	1.03
		$N = 5$	MPE	1.2037	28.8305	44.7682	0.3020	0.5680	0.4217
			ALP-Based	1.3071	28.6284	45.1704	0.3002	0.5663	0.4201
			% Diff.	8.59	0.70	0.90	0.59	0.30	0.37
Quality ladder model	High investment	$N = 3$	MPE	4.0641	23.9621	130.1799	0.5084	0.8435	0.2618
			ALP-Based	4.1093	24.0540	131.0789	0.5068	0.8427	0.2567
			% Diff.	1.11	0.38	0.69	0.31	0.09	1.94
		$N = 4$	MPE	4.8899	25.1501	149.2553	0.4090	0.7119	0.3836
			ALP-Based	5.0498	25.2980	151.2078	0.4055	0.7078	0.3683
			% Diff.	3.27	0.59	1.31	0.86	0.58	3.99
		$N = 5$	MPE	5.5733	25.9517	165.1592	0.3433	0.6103	0.5067
			ALP-Based	5.8782	26.1399	168.3259	0.3385	0.6034	0.4791
			% Diff.	5.47	0.73	1.92	1.41	1.13	5.45
	Low investment	$N = 3$	MPE	2.0424	23.4767	105.2604	0.4669	0.8152	0.2567
			ALP-Based	2.0647	23.5145	105.5504	0.4663	0.8148	0.2555
			% Diff.	1.08	0.16	0.27	0.12	0.06	0.49
		$N = 4$	MPE	2.3715	25.1526	122.4082	0.3684	0.6698	0.3825
			ALP-Based	2.4495	25.2645	123.4267	0.3663	0.6670	0.3707
			% Diff.	3.18	0.44	0.83	0.56	0.43	3.21
		$N = 5$	MPE	2.6207	26.3429	137.0153	0.3038	0.5622	0.5062
			ALP-Based	2.7742	26.4766	138.4873	0.3016	0.5581	0.4864
			% Diff.	5.53	0.50	1.06	0.75	0.74	4.07

Note: Long-run statistics computed simulating industry evolution over 10^4 periods.

often within 2.0%. The differences are similar for both the capacity competition model and the quality ladder model.

The results show that our ALP-based algorithm produces a good approximation to MPE, in instances with relatively small state spaces for which MPE can be computed. Moreover, our ALP-based algorithm requires substantially less computational effort.

Comparison with oblivious equilibrium. For large state spaces, exact MPE computation is not possible, and one must resort to approximations. In this context, we use OE as a benchmark. In an OE, each firm makes decisions based only on its own firm state and the long-run average industry state, but ignores the current industry state. For this reason, OE is much easier to compute than MPE. The main result of Weintraub, Benkard, and Van Roy, (2008) establishes conditions under

TABLE 7 Parameter Selection for Comparison with OE

	Quality Ladder	
Parameters	High Inv.	Low Inv.
θ_1	0.75	0.5
d	0.4	1.0
c	0.55	0.5
$\overline{x}$	15	15
$\tilde{\phi}$	250	250
$\tilde{\kappa}$	40	30
γ	0.1	0.1
	Capacity Competition	
q_{min}	1	
q_{max}	50	
$\overline{m}$	8	
γ	0.1	

which OE well approximates MPE asymptotically as the market size grows. Weintraub, Benkard, and Van Roy, (2010) provide an efficient simulation-based algorithm that computes a bound on the approximation error for specific models. For the purposes of our comparisons, we select parameter regimes for which OE provides accurate approximations to MPE in industries with tens of firms.

Similarly to the comparison with MPE, we consider several parameter regimes that yield different investment levels. Table 7 depicts parameter selection for each instance in this setting. We consider an additional parameter $\overline{m}$ (with default value $\overline{m} = 10$), which serves as a base market size, such that the actual market size in the industry is $m = N\overline{m}$ (so we scale the market size proportionally to the total number of firms the industry can accommodate). For these instances, computing an MPE using Algorithm 1 is infeasible for $N > 5$.

Table 8 reports the results for the different parameter regimes and models studied. There, we report OE and ALP-based long-run statistics and the percentage difference between them. Running times for these instances are on the order of hours. Although our simulation routine typically samples around 5000 relevant states, here we only consider the 1500 most visited states to keep running times relatively low. In our experience, increasing the number of sampled states (within a feasible range) does not affect significantly the proximity of the resulting economic indicators to those coming from the OE equilibrium, but it does increases memory requirement significantly. [20]

We observe that ALP-based indicators are always within 8.5% of OE indicators, and often within 4%. Because OE approximates MPE accurately in these instances, ALP-based indicators should be close to MPE indicators. The results in this section suggest that our ALP-based algorithm provides a good approximation to MPE, in instances with large numbers of firms for which OE provides a good approximation to MPE. We note that the differences in indicators in this section, although being quite small, are somewhat larger than the ones in the previous section. We believe this is partially explained by the fact that OE is also subject to some approximation error. In addition, the quantities that exhibit the larger differences (e.g., $C6$) are relatively small; hence, even though the percentage differences are larger, the absolute differences are very small.

[20] In practice, running times for large instances ($N \geq 20$) are critically determined by the amount of computational effort required to solve the linear programming subroutine. For the reported results, such linear programs had approximately 14,000 (15,000) variables and 120,000 (140,000) constraints when $N = 20$ ($N = 40$). A given equilibrium computation typically entailed solving about 40 such linear programs.

TABLE 8 **Comparison of OE and ALP-Based Indicators**

			Long-Run Statistics						
Instance		Number of Firms		Total Inv.	Prod. Surp.	Cons. Surp.	*C*6	*C*12	Entry Rate
Quality ladder	Low investment	$N = 20$	OE	7.1105	63.0029	518.1806	0.4904	0.8583	1.2534
			ALP-Based	7.1160	62.7361	509.0353	0.4676	0.8504	1.2644
			% Diff.	0.08	0.42	1.76	3.11	0.92	0.87
		$N = 40$	OE	15.2514	129.8281	1318.1610	0.2668	0.4855	2.4209
			ALP-Based	14.5708	129.3723	1283.6820	0.2473	0.4733	2.5934
			% Diff.	4.46	0.35	2.63	6.85	2.52	7.13
	High investment	$N = 20$	OE	15.1665	58.5573	613.8964	0.5224	0.8608	1.4482
			ALP-Based	15.8296	58.2390	596.9013	0.4875	0.8499	1.5112
			% Diff.	4.37	0.54	2.77	6.68	1.26	4.34
		$N = 40$	OE	31.9531	118.6970	1515.6960	0.3055	0.5262	2.8436
			ALP-Based	32.8968	118.4719	1489.1871	0.2806	0.5071	2.9826
			% Diff.	2.95	0.19	1.75	8.18	3.63	4.89
Capacity comp.		$N = 20$	OE	7.1863	67.4135	245.0830	0.6095	0.9303	1.1794
			ALP-Based	7.5839	69.0197	243.8150	0.5936	0.9200	1.1138
			% Diff.	5.53	2.38	0.52	2.61	1.10	5.55
		$N = 40$	OE	12.9678	107.5034	522.4103	0.4015	0.6831	2.5293
			ALP-Based	13.2519	109.4110	525.1520	0.4166	0.7087	2.3276
			% Diff.	2.19	1.77	0.52	3.76	3.75	7.97

Note: Long-run statistics computed simulating industry evolution over 10^4 periods.

7. Conclusions and extensions

■ The goal of this article has been to present a new method to approximate MPE in large-scale dynamic oligopoly models. The method is based on an algorithm that iterates an approximate best response operator computed via approximate mathematical programming. We provided theoretical results that justify our approach. We tested our method on a class of EP-style models and showed that it provides useful approximations for models that are of practical interest in applied economics. Our method opens the door to studying dynamics in industries which, given currently available methods, have to this point been infeasible.

In some applications, one may be interested in asymmetric equilibria in EP-style dynamic models (see, for example, Harrington et al., 2010). In this case, computational requirements are even more onerous. Of course, several details would need to be worked out, starting with our definition of state space that assumes that the identity of firms does not matter. These issues notwithstanding, we think that our general approach can be extended to compute asymmetric equilibria by modifying the best response algorithm to allow different firms (or different classes of firms) to use different strategies.

Finally, an input to our algorithm is a set of basis functions, and an important contributor to the success of our approach is the selection of good basis functions. In this article, we discuss possible ways of identifying useful sets of basis functions. Moreover, our results show that a relatively compact set of separable basis functions captures the first-order effects regarding MPE strategies in the class of models we study. There are natural extensions to this set of basis functions that may be used if a richer architecture is called for. We expect that experimentation and problem-specific knowledge can guide users of the approach in selecting effective basis functions in their applications of interest. In this way, we hope that our method will find applicability in a wide class of dynamic oligopoly models.

References

AGUIRREGABIRIA, V. AND MIRA, P., "Sequential Estimation of Dynamic Discrete Games." *Econometrica,* Vol. 75 (2007), pp. 1–54.

BAJARI, P., BENKARD, C.L., AND LEVIN, J., "Estimating Dynamic Models of Imperfect Competition." *Econometrica,* Vol. 75 (2007), pp. 1331–1370.

BERTSEKAS, D. *Dynamic Programming and Optimal Control,* Vol. 2, 2d ed. Nashua, NH: Athena Scientific, 2001.

——— AND TSITSIKLIS, J. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

BESANKO, D. AND DORASZELSKI, U., "Capacity Dynamics and Endogenous Asymmetries in Firm Size." *RAND Journal of Economics,* Vol. 35 (2004), pp. 23–49.

BOYD, S. AND VANDENBERGHE, L. *Convex Optimization,* 1st ed. New York: Cambridge, 2004.

CAPLIN, A. AND NALEBUFF, B., "Aggregation and Imperfect Competition: On the Existence of Equilibrium." *Econometrica,* Vol. 59 (1991), pp. 25–59.

DE FARIAS, D. AND VAN ROY, B., "The Linear Programming Approach to Approximate Dynamic Programming." *Operations Research,* Vol. 51 (2003), pp. 850–865.

——— AND ———. "On Constraint Sampling in the Linear Programming Approach to Approximate Dynamic Programming." *Mathematics of Operations Research,* Vol. 29 (2004), pp. 462–478.

DORASZELSKI, U., "An R&D Race with Knowledge Accumulation." *RAND Journal of Economics,* Vol. 34 (2003), pp. 19–41.

——— AND JUDD, K. "Avoiding the Curse of Dimensionality in Dynamic Stochastic Games." Working Paper, Harvard University, 2010.

——— AND PAKES, A. "A Framework for Applied Dynamic Analysis in IO." In *Handbook of Industrial Organization,* Vol. 3. Amsterdam: North-Holland, 2007.

——— AND SATTERTHWAITE, M., "Computable Markov-Perfect Industry Dynamics." *RAND Journal of Economics,* Vol. 41 (2010), pp. 215–243.

ERICSON, R. AND PAKES, A., "Markov-Perfect Industry Dynamics: A Framework for Empirical Work." *Review of Economic Studies,* Vol. 62 (1995), pp. 53–82.

FUDENBERG, D. AND TIROLE, J., *Game Theory*. Cambridge, Mass.: MIT Press, 1991.

HARRINGTON, J., ISKHAKOV, F., RUST, J., AND SCHJERNING, B. "A Dynamic Model of Leap-Frogging Investments and Bertrand Price Competition." Working Paper, University of Maryland, 2010.

JUDD, K., *Numerical Methods in Economics*. Cambridge, Mass.: MIT Press, 1998.

KRUSELL, P. AND SMITH, A.,Jr, "Income and Wealth Heterogeneity in the Macroeconomy." *Journal of Political Economy,* Vol. 106 (1998), pp. 867–896.

MASKIN, E. AND TIROLE, J., "A Theory of Dynamic Oligopoly, I and II." *Econometrica,* Vol. 56 (1988), pp. 549–570.

PAKES, A. AND MCGUIRE, P., "Computing Markov-Perfect Nash Equilibria: Numerical Implications of a Dynamic Differentiated Product Model." *RAND Journal of Economics,* Vol. 25 (1994), pp. 555–589.

——— AND ———. "Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the 'Curse' of Dimensionality." *Econometrica,* Vol. 69 (2001), pp. 1261–1281.

———,OSTROVSKY, M., AND BERRY, S., "Simple Estimators for the Parameters of Discrete Dynamic Games, with Entry/Exit Examples." *RAND Journal of Economics,* Vol. 38 (2007), pp. 373–399.

PESENDORFER, M. AND SCHMIDT-DENGLER, P., "Asymptotic Least Squares Estimators for Dynamic Games." *Review of Economic Studies,* Vol. 75 (2008), pp. 901–928.

RUST, J. "Estimation of Dynamic Structural Models: Problems and Prospects." In *Advances in Econometrics: Proceedings of the Sixth World Congress,* Vol. II. Cambridge University Press, 1994.

TRICK, M. AND ZIN, S. "A Linear Programming Approach to Solving Dynamic Programs." Working Paper, Carnegie Mellon University, 1993.

——— AND ———. "Spline Approximations to Value Functions: A Linear Programming Approach." *Macroeconomic Dynamics,* Vol. 1 (1997), pp. 255–277.

TSITSIKLIS, J. AND VAN ROY, B., "Featured-Based Methods for Large Scale Dynamic Programming." *Machine Learning,* Vol. 22 (1996), pp. 59–94.

WEINTRAUB, G., BENKARD, C., AND VAN ROY, B., "Markov Perfect Industry Dynamics with Many Firms." *Econometrica,* Vol. 76 (2008), pp. 1375–1411.

———, ———, AND ———. "Computational Methods for Oblivious Equilibrium." *Operations Research,* Vol. 58 (2010), pp. 1247–1265.