An Approximate Dynamic Programming Approach to Solving Dynamic Oligopoly Models*

Vivek Farias
MIT Sloan School

Denis Saure
University of Pittsburgh

Gabriel Y. Weintraub
Columbia Business School

July, 2011

Abstract

In this paper 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 linear 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 paper, 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 things like 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.[1] With the introduction of new estimation methods (see Pesendorfer and Schmidt-Dengler

[*] Acknowledgments: We have had very helpful conversations with Allan Collard-Wexler, Uli Doraszelski, Ariel Pakes, and Carlos Santos, 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. The third author would like to thank Lanier Benkard and Ben Van Roy for discussions that initially stimulated this project and for useful feedback. We thank the Editor Phil Haile and two anonymous referees for suggestions that greatly improved the paper. The research of the first author was supported, in part, by the Solomon Buchsbaum Research Fund. Correspondence: vivekf@mit.edu, dsaure@pitt.edu, gweintraub@columbia.edu

[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 (see Doraszelski and Pakes (2007) for an excellent survey).

(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' this computation suffers from. Note that even if it is possible to estimate the model parameters without computing an equilibrium, as in the papers 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) and 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) which is far fewer than the real world industries the analysis is directed at. Such limitations have made it difficult to construct realistic empirical models.

Thus motivated, we introduce in this paper a new method to approximate MPE in EP-style dynamic oligopoly models based on approximate dynamic programming. Our method opens up 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 grow. 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 paper 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) and de Farias and Van Roy (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 in 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) 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: (1) a quality ladder model similar to Pakes and McGuire (1994); and (2) 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), and 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' introduced by Weintraub, Benkard, and Van Roy (2008) (henceforth, OE) 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 to approximate 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 since no such alternatives are available. Nonetheless, the experience with the two aforementioned regimes suggest 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 a 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 the approximate linear programming approach.

The paper 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 linear programming approach and we discuss value function approximation; this discussion remains at a relatively conceptual level. Then, in Section 6 we provide a 'guide for practitioners' of our algorithm at a level of detail of interest to readers implementing the approach. In Section 7 we report results from computational experiments. In Section 8 we provide conclusions and discuss extensions of our work. In addition, at the end of the paper we provide several appendices with important content. Appendix A describe in detail the models we analyze. Appendices B and C develop in detail the linear programming formulation and our approximation architecture. Finally, Appendices D and E provide specifics about the theory that gives support to our approach in terms of approximation guarantees.

2 Related Literature

Our work extends the approximate linear programming approach to dynamic programming (de Farias and Van Roy (2003) and de Farias and Van Roy (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 on 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) and Weintraub, Benkard, and Van Roy (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 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) that 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 have applied the latter method for dynamic games with a low dimensional continuous state space. Trick and Zin (1993) and Trick and Zin (1997) use the linear programming approach in two-dimensional problems that arise in macroeconomics. As far as we know, our paper is the first to combine a simulation scheme to sample relevant states (a procedure inherent to the approximate linear programming approach) together with value function approximation to solve highly dimensional dynamic oligopoly models.

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

Finally, we defer more specific discussions on 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).

3.1 Model and Notation

We index time periods with nonnegative 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 $x_{it} \in \mathcal{X} = \{0, 1, 2, ..., \overline{x}\}$. 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 $\mathcal{S} = \left\{ s \in \mathbb{N}^{|\mathcal{X}|} \middle| \sum_{x=0}^{\overline{x}} s(x) \leq N \right\}$.[2] 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, i.e., there exists $\overline{\pi} < \infty$, such that $|\pi(x, s)| \leq \overline{\pi}$, for all $x \in \mathcal{X},\ s \in \mathcal{S}$.

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 $\{\kappa_{it} | t \geq 0, i \geq 1\}$ 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 $\{\zeta_{it} | t \geq 0, i \geq 1\}$ are i.i.d. and independent of $\{\kappa_{it} | t \geq 0, i \geq 1\}$. 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 make the following assumptions regarding the investment process. We assume investment is bounded, i.e., there exists a positive constant $\overline{\iota}$, such that $\iota_{it} \leq \overline{\iota},\ \forall i, \forall t$. We assume that investment is productive, i.e., $w(x, \iota, \zeta)$ is nondecreasing in ι, for all x, ζ, and that $\mathcal{P}[w(x, \iota, \zeta_{i,t+1}) > 0] > 0$, for all $\iota > 0$. Also, we assume the impact of investment on transition probabilities is continuous in the following sense: for all x, k, $\mathcal{P}[w(x, \iota, \zeta_{i,t+1}) = k]$ is continuous in ι. 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.[3] 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

[2] 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 do not matter.

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

information to the firm. We assume the random variables $\{\phi_{it} | t \geq 0\, i \geq 1\}$ are i.i.d. and independent of $\{\kappa_{it}, \zeta_{it} | t \geq 0\, i \geq 1\}$, and have a well-defined density function with support on the positive real line and finite moments. If the 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.[4] 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: (1) Each incumbent firm observes its sell-off value and then makes exit and investment decisions; (2) Each potential entrant observes its entry cost and makes entry decisions; (3) Incumbent firms compete in the spot market and receive profits; (4) Exiting firms exit and receive their sell-off values; (4) 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.

3.2 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 Appendix A.

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 firms has a probability γ of increasing its state by one level.[5] 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

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

[5] In our experiments, we eventually consider both settings where $\gamma = 0$ and $\gamma > 0$. We discuss this in more detail in Section 6.1.

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 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.

4.1 Markov Perfect Equilibrium

As a model of industry behavior we focus on pure strategy Markov perfect equilibrium (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 $\mathcal{Y} = \{(x, s) \in \mathcal{X} \times \mathcal{S} : s(x) > 0\}$[6]. 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 $\iota : \mathcal{Y} \to \mathbb{R}_+$ is an investment strategy and $\rho : \mathcal{Y} \to \mathbb{R}$ 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 $\lambda : \mathcal{S}^e \to \mathbb{R}$ 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

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

strategy is $\lambda \in \Lambda$, and the firm itself follows strategy $\mu' \in \mathcal{M}$. In particular,

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

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, $V_{\mu,\lambda}(x,s) \equiv V^{\mu}_{\mu,\lambda}(x,s)$, 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:

1. Incumbent firm strategies represent a MPE:

$$\text{(4.1)} \qquad \sup_{\mu' \in \mathcal{M}} V^{\mu'}_{\mu,\lambda}(x,s) = V_{\mu,\lambda}(x,s)\,, \quad \forall (x,s) \in \mathcal{Y}.$$

2. For all states with a positive number of potential entrants, the cut-off entry value is equal to the expected discounted value of profits of entering the industry:[7]

$$\text{(4.2)} \qquad \lambda(s) = \beta E_{\mu,\lambda}\left[V_{\mu,\lambda}(x^e, s_{t+1}) | s_t = s\right], \; \forall s \in \mathcal{S}^e.$$

Standard dynamic programming arguments establish that the supremum in part 1 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.[8]

4.2 Computation of MPE

While there are different approaches to compute 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 MPE. 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 point-wise. 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 $\mathcal{BR} : \mathcal{M} \times \Lambda \to \mathcal{M} \times \Lambda$

[7] Hence, potential entrants enter if the expected discounted profits of doing so is positive. Throughout the paper 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}).

[8] Doraszelski and Satterthwaite (2010) also provide an example of multiple equilibria in a closely related model. We note, however, that using the (approximate) best response algorithm that we introduce below, we have not been able to find two different (approximate) MPE for a given instance, even when starting from different initial conditions.

according to $\mathcal{BR}(\mu,\lambda) = (\mathcal{BR}_1(\mu,\lambda), \mathcal{BR}_2(\mu,\lambda))$, where

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

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

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

Algorithm 1 Best Response Algorithm for MPE

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

If the termination condition is satisfied with $\epsilon = 0$, we have a 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.

5 Approximate Dynamic Programming

In this section we first specialize Algorithm 1 by performing step (4) using the mathematical programming approach to dynamic programming. This method attempts to find a best response, and hence, it requires compute 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 into 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.

5.1 Mathematical Programming Approach

For some $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, consider the problem of finding a best response strategy $\mu^*_{\mu,\lambda}$; 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 $C^{\mu'}_{\mu,\lambda}$ according to:

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

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

$$C_{\mu,\lambda}V = \max_{\mu' \in \mathcal{M}} C^{\mu'}_{\mu,\lambda}V,$$

where the maximum is achieved point-wise. 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

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

where $a \vee b = \max(a,b)$ and κ is drawn according to the sell-off value distribution presumed. The best response to (μ, λ) may be found by computing a fixed point of the 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:

$$
\begin{aligned}
&\min \quad c'V \\
&\text{s.t.} \quad (T_{\mu,\lambda}V)(x,s) \le V(x,s), \quad \forall (x,s) \in \mathcal{Y}.
\end{aligned}
\tag{5.1}
$$

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:

1. **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. In Section 5.2 we show how to reduce the number of variables of the mathematical program using value function approximation.

2. **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. In Section 5.3.1 we discuss how the state relevance weight vector c plays the role of trading off approximation error across different states.

3. **Number of Constraints.** The number of constraints of the mathematical program is equal to the size of the state space. In Section 5.3.2 we introduce a constraint sampling scheme to alleviate this difficulty.

4. **Non-Linear Program.** Program (5.1) is, as stated, a non-linear program. In Section 5.4 we introduce a useful *linear* formulation of the mathematical program which is much simpler to solve than its counterpart with non-linear constraints.

After all these steps, our initial mathematical program will be transformed into a tractable linear program that computes an approximate best response. Iterating this approximate best response operator like in Algorithm 1 yields a tractable approximation to MPE. In Section 5.5 we summarize theory that provides support for our approach.

Before moving on, we digress to mention that many alternative methods to compute 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 determine 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 linear programming approach here is appealing for several reasons:

1. The crucial computational step requires the solution of a linear program which are relatively easy to solve. In contrast, collocation methods, for example, typically require solving non-linear systems of equations. In addition, we can leverage commercial linear programming software.

2. 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.

5.2 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:

$$\begin{array}{ll} \min & c'\Phi r \\ \text{s.t.} & (T_{\mu,\lambda}\Phi r)(x,s) \leq (\Phi r)(x,s) \quad , \forall (x,s) \in \mathcal{Y}. \end{array} \tag{5.2}$$

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 (5.2) 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 Appendices D.1 and D.2.

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.

5.2.1 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. At the end of the section 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 $\mathcal{X} = \{0, ..., 3\}$, 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:

- 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 $\mathcal{Y}$).

- 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 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

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

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 $(x, s(0), s(1))$, while this approximation only allows for the separably additive form.

- 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:

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

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; see Appendix B for details.

Evidently, by picking appropriate sets $\mathcal{C}_x$, the basis functions described can be used to capture a rich array of approximations. For example, one could also 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:

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

Of course, this specification requires a larger number of basis functions.

Selecting an appropriate set of basis functions is problem dependent. In what follows, we demonstrate numerical examples with the general separable, and moment based approximation architectures.

5.2.2 Examples

We test the previously introduced approximation architectures in the quality ladder and capacity models introduced in Section 3.2 and described in detail in Appendix A. 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.

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 low 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 the different set of basis functions and the exact MPE. Except for a boundary effect at the firms 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 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 yield an R^2 of 0.9 and 0.97, respectively; the impact of the competitors' state in equilibrium outcomes can indeed be summarized by 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

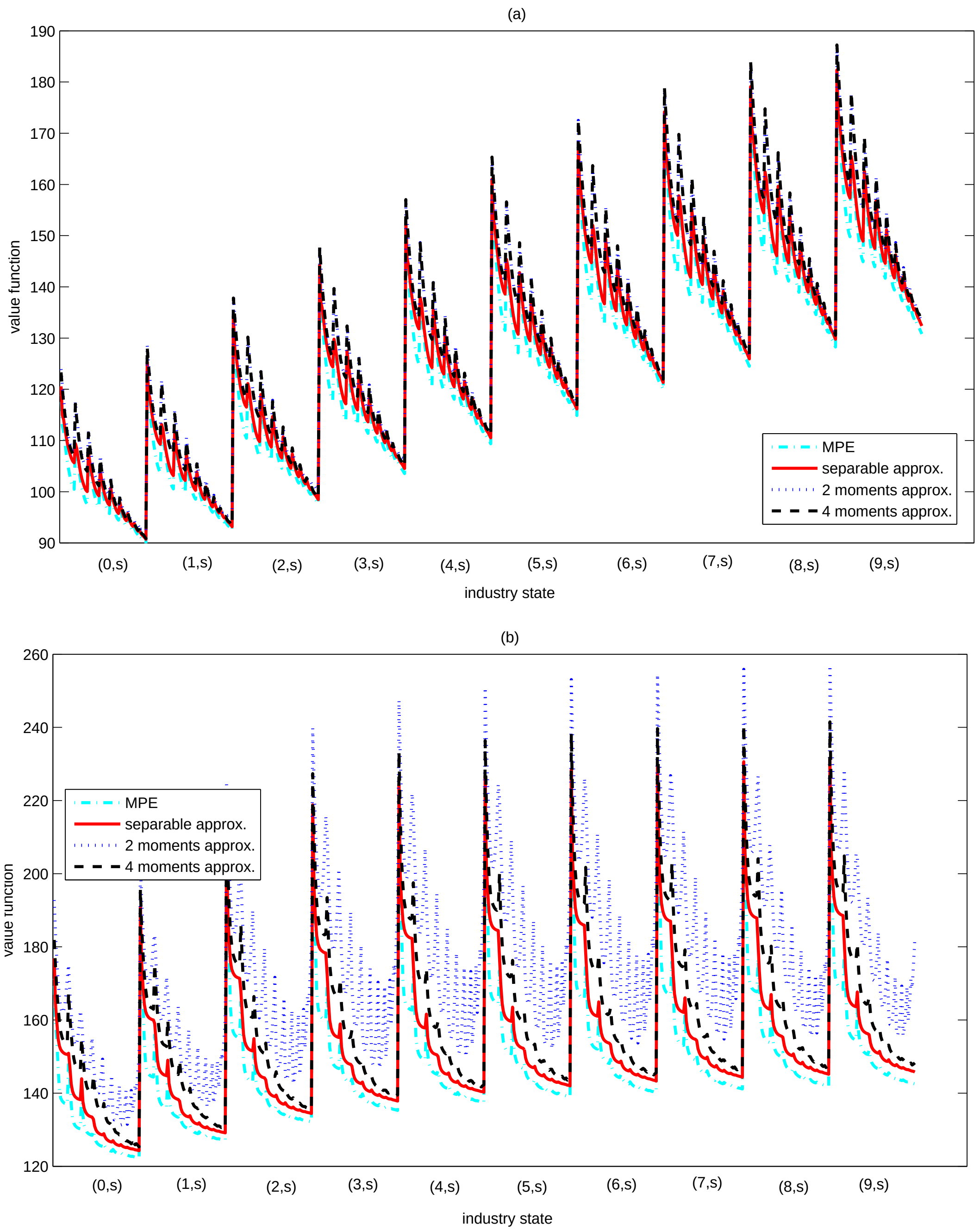

Figure 1: Value function approximation for different sets of basis functions and competition models. The graph in (a) compares approximations from separable and moment-based architectures for the quality ladder model. The graph in (b) compares approximations from separable and moment-based architectures 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.

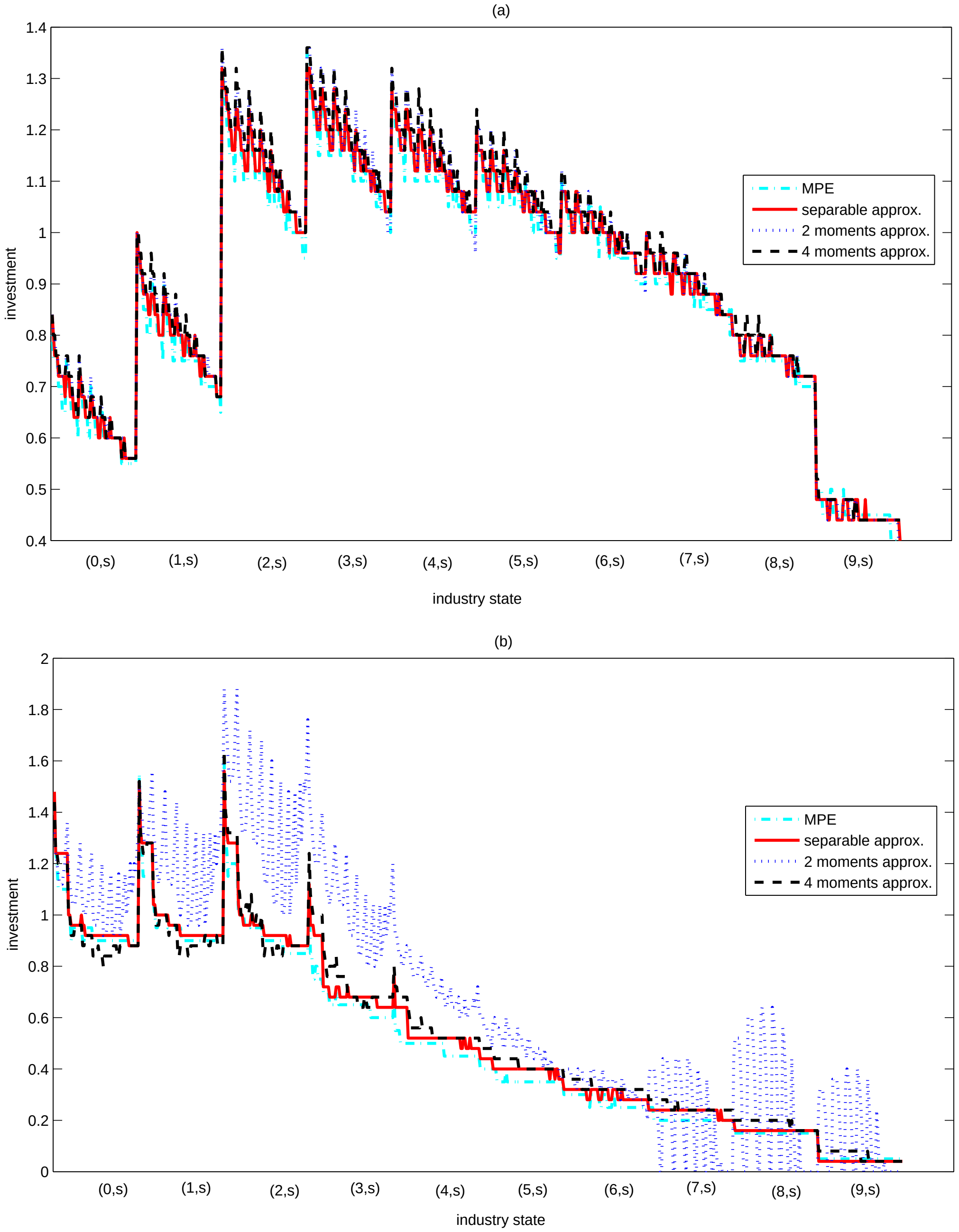

Figure 2: Investment strategy approximation for different sets of basis functions and competition models. The graph in (a) compares strategies from separable and moment-based architectures for the quality ladder model. The graph in (b) compares strategies from separable and moment-based architectures 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.

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 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 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.

5.2.3 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 7 can support this process in practice. In Section 7 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 paper 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 provide more flexibility and are 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.

5.3 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.

5.3.1 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 Section D.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 like 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. 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.

5.3.2 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 (5.1). 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 since the number of variables common to all constraints in (5.2) is small i.e. 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 (5.2):

$$\begin{array}{ll} \min & c'\Phi r \\ \text{s.t.} & (T_{\mu,\lambda}\Phi r)(x,s) \leq (\Phi r)(x,s) \quad \forall (x,s) \in \mathcal{R}. \end{array} \tag{5.3}$$

Intuitively, a sufficiently large number of samples L should suffice to guarantee that a solution to (5.3) satisfies all except a small set of constraints in (5.2) 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 (5.3) whose quality is similar to that of an approximate best response computed via the intractable program (5.2). We illustrate this point with an example and present theoretical support in Appendix D.3.

Earlier, we pointed 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 3.2 confirmed that, compared to other alternatives, the selections we suggest consistently provide the best approximations to the MPE computed with our best response algorithm.

5.3.3 Examples

Consider the capacity competition model introduced in Section 3.2 and described in detail in Appendix A, 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 (5.3) for several values of $|\mathcal{R}|$, while considering (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 (5.3) 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 (5.3) are $\mathcal{R} = \mathcal{Y}_{LR}(r)$ for $r = 500, 250, 100, |\mathcal{Y}|$. With these we compute the following weighted approximation relative errors:

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

The errors above quantify the quality of the approximation produced while 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 (5.3) 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.[10]

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

Table 1: Weighted relative errors of value function for capacity competition model, using moment-based approximation with two moments when $N = 4$ and $c = c_{LR}$.

5.4 Discretization and a Tractable Linear Program

In solving the program (5.3), it is computationally challenging to allow for a continuum of investment levels and for sell-off values to be continuous random variables. As such, by suitably discretizing allowable investment levels and by approximating a continuous valued random variable by a discrete random variable, we may hope to approximate the solution of the original, continuous problem. Here we provide the details of this discretization.

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 in the set $\hat{\mathcal{K}} = \{k_1, k_2, \ldots, k_n\}$ where k_j as the largest quantity satisfying $\mathcal{P}[\kappa < k_j] \le \frac{n+1-j}{n}$ for $j = 1, \ldots, n$. 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

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

[10] 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.

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

The above discretizations are tantamount to the approximation

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

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 (5.3):

$$\begin{aligned} &\min \quad c'\Phi r \\ &\text{s.t.} \quad (T^{\epsilon,n}_{\mu,\lambda}\Phi r)(x,s) \leq (\Phi r)(x,s), \quad \forall (x,s) \in \mathcal{R}. \end{aligned} \tag{5.4}$$

Two questions arise regarding the program (5.4). First, what impact does discretization have on our accuracy in solving the best response problem? Second, why is (5.4) any easier to solve? The first question is answered in Appendix C, where we show that the approximation does not degrade much with a fine enough discretization of the sell-off value distribution and investment levels. In addition, computational experimentation with the models presented in Section 3.2 confirm this. As for the second question, (5.4) is, in fact, equivalent to a linear program which is substantially simpler to solve than the non-linear program (5.3); this equivalent linear program is described next.

Notice that by introducing auxiliary variables $u(x,s) \in \mathbb{R}^n$, the constraint $(T^{\epsilon,n}_{\mu,\lambda}V)(x,s) \leq V(x,s)$ is equivalent to the following set of constraints:[11]

$$\begin{aligned} &\pi(x,s) + \tfrac{1}{n}\textstyle\sum_{i=1}^{n} u(x,s)_i \leq V(x,s) \\ &\max_{\mu'(x,s):\iota'(x,s)\in\mathcal{I}^{\epsilon}} C^{\mu'}_{\mu,\lambda}V(x,s) \leq u(x,s)_i \quad \forall i \in \{1,...,n\} \\ &k_i \leq u(x,s)_i \quad \forall i \in \{1,...,n\}. \end{aligned} \tag{5.5}$$

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:

$$-d\iota' + \beta E_{\mu,\lambda}[V(x_1,s_1)|x_0 = x, s_0 = s, \iota_0 = \iota'] \leq u(x,s)_i \ , \ \forall \iota' \in \mathcal{I}^{\epsilon}.$$[12]

Hence, (5.4) is, in fact equivalent to the following linear program:

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

[11] By equivalent we mean that the set values of $V(x,s)$ that satisfy the constraint is identical to the set of values of $V(x,s)$ that satisfy (5.5).

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

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 solve this program is the expectation over next period states that we need to compute in the left hand side of the constraints in (5.4); as pointed out by Doraszelski and Judd (2010) this involves a high dimensional sum. We will show later, however, that this sum gets significantly simplified with the basis functions we use.

One last comment is at order. Introducing discrete investment levels and sell-off values may destroy the existence of pure strategy equilibrium as shown by Doraszelski and Satterthwaite (2010). We do not allow for mixed strategies. Instead, when computing an MPE we relax the stopping criteria in Algorithm 1 as we now explain. For simplicity consider a model with investment decisions only. Given our assumption that the investment process is unique investment choice admissible, 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, we stop and we consider $(\iota_i + \iota_{i+1})/2$ as an approximation to a pure strategy MPE. It is easily seen that as ϵ becomes small, we get closer to a pure strategy MPE of the continuos model. A similar argument would apply for discretizing sell-off values, however our algorithm considers an alternative approach that does not require such discretization; see Section 6.2.1.

In Section 6 we will provide a procedural description of Algorithm 1 wherein the best response computation step is approximated and accomplished using the tractable program (5.4). That section will present the overall scheme at a level of detail of interest to readers implementing the approach. Before that, we conclude this section by briefly describing the theoretical guarantees we can offer for our approach.

5.5 Theoretical Guarantees

In Sections D and E in the Appendix we introduce theory that provides support for our overall approach. We summarize those theoretical guarantees here.

We first develop an extension of the theory developed in de Farias and Van Roy (2003) and de Farias and Van Roy (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 $\epsilon-$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 $\epsilon-$equilibrium and approximating equilibrium strategies that provides a more direct test of the accuracy of our approximation. In Theorem E.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 a MPE. The result is valid for general approximations techniques and we anticipate it can be useful to justify other approximation schemes for dynamic oligopoly models or even in other contexts.

6 A Procedural Description of the Algorithm

This section is a procedural counterpart to the preceding section. In particular, we provide a procedural description of the linear programming approach to compute an approximate best response in lieu of step (4) in Algorithm 1. The overall procedure is described as Algorithm 2 in Section 6.1. The following sections present important sub-routines.

6.1 Algorithm

Our overall algorithm employing the approximate best response computation procedure, Algorithm 2, will require the following inputs:

1. $\{\Phi_i \,:\, i = 1, \ldots, K\}$, a collection of K basis functions. This collection is such that $\Phi_i : \mathcal{Y} \rightarrow \mathbb{R}$ for all $i = 1, 2, \ldots, K$. We denote by $\Phi \in \mathbb{R}^{|\mathcal{Y}| \times K}$ the matrix $[\Phi_1, \Phi_2, \ldots, \Phi_K]$.

2. A discrete valued random variable $\hat{\kappa}$ taking values uniformly at random in a set $\{k_i \,:\, i \in \hat{\mathcal{K}}\}$ where $\hat{\mathcal{K}}$ is a finite index set with cardinality n. Such a random variable may be viewed as a discretization to a given sell-off value distribution as described in the previous section. It will also be possible to deal with continuous random variables; see Section 6.2.1.

3. A discrete set of permissible investment levels $\mathcal{I} = \{0, \epsilon, 2\epsilon, \ldots, \lfloor \bar{\iota}/\epsilon \rfloor \epsilon\}$. Again, $\mathcal{I}$ may be viewed as a discretization of some given set of permissible investment levels.

4. (μ^c, λ^c), an initial investment/exit strategy and entry cut-off rule with a compact representation. An example of such a strategy derived from what is essentially a myopic strategy is given by:

$$\begin{array}{rll} \iota^c(x,s) & = 0, & \forall (x,s) \in \mathcal{Y} \\ \rho^c(x,s) & = \frac{1}{1-\beta} \, \mathrm{E}[\pi(x_1, s) | x_0 = x \,,\, \iota = \iota^c], & \forall (x,s) \in \mathcal{Y} \\ \lambda^c(s) & = \frac{1}{1-\beta} \pi(x^e, s), & \forall s \in \mathcal{S}^e \end{array}$$

5. An arbitrary initial state in $\mathcal{Y}$, v.

6. Tuning parameters: $i)$ $\tilde{L}$, a positive integer required to calibrate simulation effort and size of the linear program to solve at each iteration (see step (4) in Algorithm 2); $ii)$ T, a positive integer that determines the size of the transient period when simulating the industry evolution; $iii)$ $\epsilon > 0$ used to

calibrate the stopping criteria (see step (13) in Algorithm 2); and $iv)$ γ_{simul}, an appreciation factor used in constructing distributions to sample industry states.

We next describe our Algorithm, noting that the description will call on two procedures we are yet to describe: (1) an approximate linear programming sub-routine $ALP(\cdot)$, and (2) an oracle $M(\cdot)$ that succinctly represents investment, entry and exit strategies using the results of previous iterations and is called whenever the incumbent strategy in a given state needs to be computed. The next two sections are devoted to describing these sub-routines in detail.

Upon convergence, the approximate MPE strategy can be recovered as $(\mu_{i^*}, \lambda_{i^*}) = M(r^{i^*}, \ldots, r^1, \mu^c, \lambda^c)$, where i^* is the number of iterations until convergence, and r^j are the weights obtained in $ALP(\cdot)$ at each iteration. There are several aspects of Algorithm 2 that merit further comment:

1. **Sampling States:** The ADP scheme relies on samples drawn from the state space under the incumbent strategy in the algorithm. In particular, this is done in step (4) of Algorithm 2 and the samples obtained are used for constructing c and $\mathcal{R}$. The tuning parameters v and T are chosen appropriately. For example, if one is interested in approximating the long-run behavior of the industry, T is set to be a very large number. In contrast, if one is interested in approximating the short-run behavior of the industry starting from a given initial state, then $T = 0$ and v is set to be the initial state of interest.

   We note here that the distribution used to sample states can assume a somewhat distinct model of industry dynamics than the true model. In particular, if the model appreciation parameter $\gamma = 0$, we may sample assuming industry dynamics with an appreciation parameter $\gamma_{\text{simul}} > 0$. We do this as a means of encouraging 'exploration' when an intermediate policy spends most of its time in a small part of the state space. If $\gamma > 0$, we take $\gamma_{\text{simul}} = \gamma$, so that in this case the dynamics assumed by the distribution used to sample states coincides with the actual dynamics. In our experience, we observed that sometimes setting a small but positive value for the model parameter γ improved the approximation.

2. **Convergence and Stopping Criteria:** Our algorithm stops when a proxy to the expected benefit a firm obtains from unilaterally deviating from the approximate policy, Δ, falls below a specified threshold, ϵ. In the ith iteration of the algorithm, this proxy is defined according to $\Delta_i = \sum_{(x,s)\in\mathcal{R}} c(x,s) \left| V^{\mu_{i+1}}_{\mu_i \lambda_i}(x,s) - V_{\mu_i \lambda_i}(x,s) \right|$ and evaluated via simulation in step (11) of the algorithm. In essence, this treats μ_{i+1} as a proxy for the best response to (μ_i, λ_i). Note that the empirical distribution c is computed using the incumbent strategies in the algorithm, (μ_i, λ_i), and therefore changes in each iteration. Appendix E relates convergence of the algorithm under this proxy to convergence to an MPE strategy.

3. **Smooth Updates:** In our computational experiments we allow for a 'smooth' update of the r values. Specifically we performed the update $r_i := r_{i-1} + (r_i - r_{i-1})/i^{\varrho}$ as a last step in every iteration. The

Algorithm 2 Algorithm for Approximating MPE

```
$\mu_0 = \mu^c$, $\lambda_0 = \lambda^c$
{Set initial investment, entry and exit strategies}
$i := 0$
{$i$ indexes best response iterations}
repeat
    Simulate industry evolution over $T + \tilde{L}$ periods, assuming all firms use strategy $(\mu_i, \lambda_i)$, the initial industry state is $v$, and investment appreciates according to $\gamma_{simul}$.
      • Compute empirical distribution $c$.
        $c(x,s) = \sum_{t=T}^{T+\tilde{L}} \mathbf{1}[s_t(x) > 0\,,\ s_t = s] / \sum_{t=T}^{T+\tilde{L}} \sum_{x \in \mathcal{X}} \mathbf{1}[s_t(x) > 0]\ ,\ \forall (x,s) \in \mathcal{Y}$
      • $\mathcal{R} \leftarrow \{(x,s) \in \mathcal{Y} \,:\, c(x,s) > 0\}$
      • Let $L = |\mathcal{R}|$.
    {The distribution $c$ is the empirical counterpart of the set of "relevant" states induced by $(\mu_i, \lambda_i)$. $c$ and $\mathcal{R}$ are used to build the ALP; see Appendix D for the theoretical justification.}
    Set $r^{i+1} \leftarrow ALP_\theta(\mathcal{R}, \mu_i, \lambda_i, c)$
    {The $ALP_\theta(\cdot)$ procedure produces an approximate best response to $(\mu_i, \lambda_i)$; this is succinctly described by the parameter vector $r^{i+1}$. See the following Section for the description of $ALP_\theta(\cdot)$; $\theta$ is a regularization parameter for the procedure.}
    $(\mu_{i+1}, \lambda_{i+1}) := M(r^{i+1}, \ldots, r^1, \mu_0, \lambda_0)$.
    {The oracle $M(\cdot)$ described in Section 6.3 uses the weight vectors $r^i$ to generate the corresponding investment, entry and exit strategies at any given queried state; this does not require an explicit description of those strategies over the entire state space which is not tractable.}
    for each $(x,s) \in \mathcal{R}$ do
        Estimate $V_{\mu_i \lambda_i}^{\mu_{i+1}}(x,s)$.
        {Estimation is via monte-carlo simulation of industry evolution starting from state $(x,s)$ with the incumbent firm using strategy $\mu_{i+1}$ and its competitors using $(\mu_i, \lambda_i)$.}
        Estimate $V_{\mu_i \lambda_i}(x,s)$.
        {Estimation is via monte-carlo simulation of industry evolution starting from state $(x,s)$ with all firms using strategy $(\mu_i, \lambda_i)$.}
    end for
    $\Delta = \sum_{(x,s) \in \mathcal{R}} c(x,s) \left| V_{\mu_i \lambda_i}^{\mu_{i+1}}(x,s) - V_{\mu_i \lambda_i}(x,s) \right|$
    {$\Delta$ is the empirical counterpart to the directly measurable component of the quality of a candidate equilibrium; see Appendix E.}
    $i := i + 1$
until $\Delta < \epsilon$
```

parameter ϱ was set after some experimentation equal to $2/3$. We observe from our experiments that using an update of this sort was beneficial to the rate of convergence of our scheme.

6.2 The Linear Programming Sub-routine $ALP_\theta(\cdot)$

The $ALP_\theta(\mathcal{R}, \mu, \lambda, c)$ sub-routine employed in step (5) of Algorithm 2 simply outputs the solution of the following linear program:

(6.1)

$$\begin{array}{lll}
\underset{r,t,u,l}{\text{minimize}} & \displaystyle\sum_{(x,s)\in\mathcal{R}} c(x,s) \sum_{0\le j\le K} \Phi_j(x,s)\, r_j & \\
\text{subject to} & \displaystyle\pi(x,s) + \frac{1}{n}\sum_{i=1}^{n} u(x,s)_i \le \sum_{0\le j\le K} \Phi_j(x,s)\, r_j + l(x,s) & \forall (x,s)\in\mathcal{R} \\
& \displaystyle -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0\le j\le K} \Phi_j(x_1,s_1)\, r_j \middle| x_0 = x, s_0 = s, \iota_0 = \iota\right] \le t(x,s) & \forall (x,s)\in\mathcal{R}, \iota\in\mathcal{I} \\
& t(x,s) \le u(x,s)_i & \forall (x,s)\in\mathcal{R}, i\in\hat{\mathcal{K}} \\
& k_i \le u(x,s)_i & \forall (x,s)\in\mathcal{R}, i\in\hat{\mathcal{K}} \\
& \frac{1}{|\mathcal{R}|}\sum_{(x,s)\in\mathcal{R}} l(x,s) \le \theta & \\
& l(x,s) \ge 0 & \forall (x,s)\in\mathcal{R}
\end{array}$$

To prevent this program from being unbounded we also included the constraint that r lies in a large bounded set. When the parameter θ is set to 0, the above linear program (LP) is, in fact, equivalent to the LP derived for the program (5.4) [13] Briefly, we recall that we expect that given an optimal solution r to the above program, Φr should provide as good an approximation to $V^*_{\mu,\lambda}$ as is possible with the approximation architecture Φ. Positive values of θ serve to regularize the program by allowing violations of the Bellman inequalities in states where this may benefit the overall approximation; the theory supporting this regularization is developed in Desai, Farias, and Moallemi (2010). In particular, that paper extends the theory presented in Appendix D to the "regularized" LP. Desai, Farias, and Moallemi (2010) provide a theoretically robust choice of θ. In our own numerical experiments we determine an ideal choice of θ in small instances where exact MPE is computable and employ this choice in our experiments with a large number of firms. We first tried $\theta = 0$ and then explore positive values of θ if required. In practice, we used both $\theta = 0$ and $\theta > 0$ depending on the instance.

The second set of constraints in (6.1) involve the expectations $E_{\mu,\lambda}\left[(\Phi r)(x_1, s_1)|x_0 = x, s_0 = s, \iota_0 = \iota\right]$. For both the quality ladder and quantity competition models, the separable architecture introduced in Section 5.2.1 allows us to express these expectations in a tractable fashion. In particular, these expectations can be written as linear functions in r whose coefficients may be computed with roughly $|\mathcal{X}|N^4$ operations.[14]

[13] With the only difference that here we have introduced the auxiliary variables $t(x,s), \forall (x,s) \in \mathcal{R}$, to reduce the number of constraints of the program. However, the programs are indeed equivalent.

[14] In that model, firms can only transition to adjacent individual states. Considering this and the nature of the basis functions,

This may not be possible in other models or with other types of basis functions; in that case one may simply replace the expectation with its empirical counterpart. Alternatively, Doraszelski and Judd (2010) propose a continuous time formulation that significantly reduces the complexity involved in computing the expectation over next period states.

Problem (6.1) has $(K + L(n + 2))$ decision variables and $(L(|\mathcal{I}| + 2 \cdot n + 2) + 1)$ constraints. Thus both the number of constraints and variables in (6.1) scale proportionally to the number of states L sampled from simulating the industry evolution. Note that the number of variables and constraints do not scale with the size of the state space and one may solve this LP to optimality directly provided L is sufficiently small. However, an alternative procedure proved to provide a further speedup. We describe this procedure next.

6.2.1 A Fast Heuristic LP Solver

We present here a fast iterative heuristic for the solution of the LP (6.1) that constitutes the subroutine $ALP_\theta(\mathcal{R}, \mu, \lambda, c)$. As opposed to solving a single LP with $(K + L(n+2))$ decision variables and $(L(|\mathcal{I}| + 2 \cdot n + 2) + 1)$ constraints as above, the procedure solves a sequence of substantially smaller LPs, each with $(K + L)$ variables and $(L(|\mathcal{I}| + 1) + 1)$ constraints. The idea underlying the heuristic is quite simple: whereas LP (6.1) essentially attempts to find an optimal investment strategy and exit rule in response to some input strategy (μ, λ), the heuristic assumes a fixed exit rule, and attempts to find a near optimal investment strategy; following this, the exit rule is updated to reflect this new investment strategy and one then iterates to find a new near optimal investment rule given the updated exit rule. As we will show, any fixed point of this procedure is indeed an optimal solution to the LP (6.1). First, we describe the heuristic in detail; see Algorithm 3.

Algorithm 3 stops when the exit strategy implied by the current optimal investment levels are consistent with the exit strategy from which those investment levels where derived in the first place. The fixed points of the above approach constitute an optimal solution to (6.1). In particular, suppose $\epsilon = 0$ in the specification of Algorithm 3 and let r' and l' be the output of the Algorithm assuming it terminates. Define

$$t'(x,s) = \max_{\iota \in \mathcal{I}} -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0 \le k \le K} \Phi_k(x_1, s_1)\, r'_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota\right], \quad \forall (x,s) \in \mathcal{R}.$$

and

$$u'(x,s)_i = \max\{t'(x,s), k_i\}, \quad \forall (x,s) \in \mathcal{R}, i \in \hat{\mathcal{K}}.$$

We then have the following result that we prove in the Appendix.

Proposition 6.1. *(r', u', t', l') is an optimal solution to the LP* (6.1).

given a state (x, s) it is enough to go over each possible individual state $j \in \mathcal{X}$ and compute the probability distribution of the number of firms that will transition to state j from states $j, j-1$, and $j+1$, while at the same time considering firms leaving/entering the industry.

Algorithm 3 Heuristic to solve Linear Program $ALP_\theta(\mathcal{R}, \mu, \lambda, c)$

```
1: $j := 0$
2: $r' = r$
   {$r$ could be arbitrary here; a useful initial condition is to consider $r$ to be the value computed at the previous best response iteration.}
3: Set $e_j(x,s) = \max_{\iota \in \mathcal{I}} -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0 \le k \le K} \Phi_k(x_1, s_1)\, r'_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota\right], \forall (x,s) \in \mathcal{R}$.
   {Set cutoff values for firm exit based on the current approximation to the optimal value function.}
4: repeat
5:     Set $r'$ and $l'$ as a solution to
       (6.2)
       $\underset{r,l}{\text{minimize}} \quad \sum_{(x,s) \in \mathcal{R}} c(x,s) \sum_{0 \le k \le K} \Phi_k(x,s)\, r_k$
       subject to $\quad \pi(x,s) + \mathcal{P}(\hat{\kappa} < e_j(x,s)) \left( -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0 \le k \le K} \Phi_k(x_1, s_1)\, r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota\right]\right)$
           $+E[\hat{\kappa} | \hat{\kappa} \ge e_j(x,s)]\mathcal{P}(\hat{\kappa} \ge e_j(x,s)) \le \sum_{0 \le k \le K} \Phi_k(x,s)\, r_k + l(x,s)$ ,
           $\forall (x,s) \in \mathcal{R}, \iota \in \mathcal{I}$,
           $\frac{1}{|\mathcal{R}|} \sum_{(x,s) \in \mathcal{R}} l(x,s) \le \theta$,
           $l(x,s) \ge 0 \qquad \forall (x,s) \in \mathcal{R}$.
       {Compute an approximate best response investment strategy assuming the *fixed* exit rule determined by the cutoff value $e_j$.}
6:     Set $e_{j+1}(x,s) = \max_{\iota \in \mathcal{I}} -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0 \le k \le K} \Phi_k(x_1, s_1)\, r'_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota\right], \forall (x,s) \in \mathcal{R}$.
       {Update the exit rule based on the computed approximate best response investment strategy.}
7:     $\Delta := \sum_{(x,s) \in \mathcal{R}} c(x,s)|e_{j+1}(x,s) - e_j(x,s)|$
       {If the update in the exit rule is sufficiently small then the computed value function in step (5) is, in fact, a near-optimal solution to (6.1).}
8:     $j := j + 1$
9: until $\Delta < \epsilon$
10: return $r'$ and $l'$
```

It is not clear that Algorithm 3 is convergent. Note however that if the algorithm did not converge within a user specified number of iterations, one can always resort to solving (6.1) directly. In practice, the number of iterations required for convergence depends on how close (μ_i, λ_i) is to an approximate equilibrium: when Algorithm 2 is close to convergence we expect the number of inner iterations in Algorithm 3 to be very small. Also, during the initial few iterations of Algorithm 2 one can restrict the number of inner iterations in Algorithm 3; in the initial steps of the algorithm when presumably the strategies are not close to an approximate equilibrium, having an accurate approximation to a best response is not crucial. In practice, using this scheme (as opposed to solving (6.1) directly) provided an average speedup of one order of magnitude.[15]

It is also worth remarking that the above procedure does not really require that the selloff value distribution be discrete. In fact, we use the actual (exponential) sell off value distribution in our experiments without the discretization. In the event that the procedure above converges, one could show that the resulting solution is in fact an optimal solution to the program (5.3) (assuming discrete investment levels); the proof essentially follows that of the above proposition.

We next describe the sub-routine $M(\cdot)$ which we recall serves as an oracle procedure for the computation of the current investment strategy μ and entry rule λ at a given input state.

6.3 Computing Strategies Given a Sequence of Weight Vectors: The Oracle M

At several points in Algorithm 2, we require access to the current candidate equilibrium strategy (μ_i, λ_i). More precisely, we require access to a procedure that given a state $(x, s) \in \mathcal{Y}$ or a state $s \in \mathcal{S}^e$ efficiently computes $\mu_i(x, s)$ or $\lambda_i(s)$, respectively, at any stage i of the algorithm. Simply storing (μ_i, λ_i) in a look-up table is infeasible given the size of $\mathcal{Y}$. Fortunately, we can develop a sub-routine that given past approximate best-responses (encoded via the weight vectors r^i), an initial strategy with a compact representation, and an input state $(x, s) \in \mathcal{Y}$ $(s \in \mathcal{S}^e)$, is able to efficiently generate $\mu_i(x, s)$ $(\lambda_i(s))$. We specify this sub-routine, M, in this section. We will show that $M(\cdot)$ runs in time that is only linear in the current iteration count i.

Fix $(x, s) \in \mathcal{Y}$, and define $\mathfrak{N}(x, s)$ as the set of possible states faced by firms in that industry state, i.e,

$$\mathfrak{N}(x, s) = \{(y, s) \in \mathcal{Y} : s(y) > 0\}.$$

For any given state $(x, s) \in \mathcal{Y}$, Algorithm 4, described below, computes $(\mu_i(x, s), \lambda_i(s))$ using as input the sequence of previous solutions to (6.1) and the initial compact-representation strategy (μ^c, λ^c) (it is understood that only $\mu_i(x, s)$ is computed when $s \notin S^e$).

Next, we argue the complexity of Algorithm 4 increases linearly with i, the current iteration count at which a call to $M(\cdot)$ is made in Algorithm 2. For that we need the following key observations:

[15] In practice, the number of simplex iterations required to solve an LP typically grows linearly in the number of constraints; see Applegate, Bixby, Chvatal, and Cook (2006). Our scheme trades solving a handful of LPs for reducing the number of constraints of each LP by a factor of n

Algorithm 4 $M(r^i, r^{i-1}, \dots, r_1, \mu^c, \lambda^c)$ (Computation of $(\mu_i(x,s), \lambda_i(s))$ for $(x,s) \in \mathcal{Y}$)

```
1: $\mu_0 := \mu^c$, $\lambda_0 := \lambda^c$, $j := 1$
2: repeat
3:     for all $(y,s) \in \mathfrak{N}(x,s)$ do
4:
           $$\iota_j(y,s) := \operatorname*{argmax}_{\iota \in \mathcal{I}} \; -d\iota + \beta E_{(\mu_{j-1},\lambda_{j-1})}\left[\sum_{0\le k\le K} \Phi_k(x_1, s_1) r_k^j \Big| x_0 = y, s_0 = s, \iota_0 = \iota\right].$$
           $$\rho_j(y,s) := -d\iota_j(y,s) + \beta E_{(\mu_{j-1},\lambda_{j-1})}\left[\sum_{0\le k\le K} \Phi_k(x_1, s_1) r_k^j \Big| x_0 = y, s_0 = s, \iota_0 = \iota_j(y,s)\right].$$
           $$\lambda_j(s) := \beta E_{(\mu_{j-1},\lambda_{j-1})}\left[\sum_{0\le k\le K} \Phi_k(x^e, s_1) r_k^j \Big| s_0 = s\right].$$
5:         $j := j + 1$
6:     end for
7: until $j = i$
8: return $\iota_i(x,s), \rho_i(x,s), \lambda_i(s)$
```

- For all $(x,s) \in \mathcal{Y}$, we have that $\mathfrak{N}(y,s) = \mathfrak{N}(x,s)$, for all $(y,s) \in \mathfrak{N}(x,s)$.

- For all $(x,s) \in \mathcal{Y}$, we have that $|\mathfrak{N}(x,s)| \le \min\{N, |\mathcal{X}|\}$.

Note that in the last iteration of Algorithm 4 in steps (4) to (6) we require, in addition to knowing r^j, which is easy to store, that we compute $(\mu_{j-1}(y,s), \lambda_{j-1}(s))$ for states $(y,s) \in \mathfrak{N}(x,s)$. Unless $j-1 = 0$, computing $(\mu_{j-1}, \lambda_{j-1})$ will in turn require knowledge of r^{j-1} and $(\mu_{j-2}, \lambda_{j-2})$ for states $(y', s) \in \mathfrak{N}(y,s)$. Since $\mathfrak{N}(y,s) = \mathfrak{N}(x,s)$, $\forall y$, we note that the set of states we must compute actions for at each level of the recursion is always contained in $\mathfrak{N}(x,s)$. This fact, depicted in Figure 3, prevents the computation from blowing up; since $|\mathfrak{N}(x,s)| \le \min\{N, |\mathcal{X}|\}$, Algorithm 4 makes no more than $i \cdot \min\{N, |\mathcal{X}|\}$ calls to line 4 in computing $(\mu_i(x,s), \lambda_i(s))$. Hence, the computational effort increases only linearly in the number of iterations. An alternative to the oracle $M(\cdot)$ can be developed for which the computational effort does not increase with the number of iterations. This formulation requires $Q-$functions (Bertsekas and Tsitsiklis 1996) and the solution of an alternative ALP that demands a more complex approximation architecture. For this reason, we use the oracle $M(\cdot)$ instead.

7 Computational Experiments

In this section we conduct computational experiments to evaluate the performance of our algorithm in situations where either we can compute a MPE, or a good approximation is available. We use both the quality

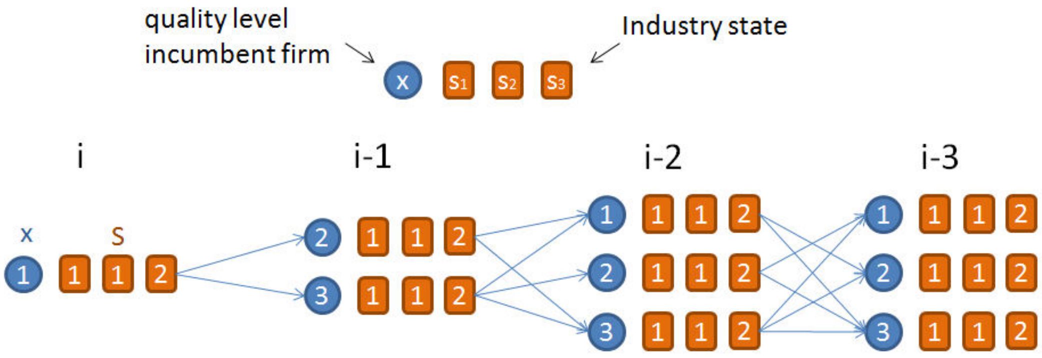

Figure 3: **Oracle computation increases linearly with the number of iterates in Algorithm 2.** Suppose we want to compute $\mu_i(x, s)$ and that $x = 1$ and $s = (1, 1, 2)$ (here $|\mathcal{X}| = 3$), that is, the industry state consists of one firm at individual states 1 and 2, and 2 firms at individual state 3; the incumbent firm under consideration is in individual state 1. Computation of $\mu_i(x, s)$ requires μ_{i-1} for the competitors of the firm in individual state 1, that is, for firms in states $(2, s)$ and $(3, s)$, where $s = (1, 1, 2)$ as before. In turn, $\mu_{i-1}(2, s)$ requires μ_{i-2} for firms in states $(1, s)$ and $(3, s)$. We can continue with this reasoning and observe that all states for which we need to recover strategies in this iterative scheme are contained in $\mathfrak{N}(x, s)$.

ladder and the capacity competition models described in Section 3.2.[16]

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 5.2.1. 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 $|\mathcal{Y}|$, hence the use of this approximation architecture makes the linear program in Algorithm 3 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 the linear program in Algorithm 3 as follows: For a set $\mathcal{H} \subseteq \{0, ..., N\}$ 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:

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

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

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

7.1 Comparing Economic Indicators of Interest

We show that our approximate linear programming-based (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 (OE) introduced by Weintraub, Benkard, and Van Roy (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, 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 $C2$, while if $N = 40$ one may also be interested in $C6$).[17]

7.1.1 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 a 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 2 depicts parameter selection for each instance.[18]

Table 3 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]

[17] Since the outcome of our algorithm is random, due to the sampling of constraints, ALP-based quantities reported in this subsection represent the average of 5 runs. On each run, industry evolution is simulated during 10^4 periods. The resulting sample of 5 data points is such that for each indicator, the ratio between the sample standard deviation and the sample mean is in average less than 1% and always less than 7%.

[18] We also note that in our computational experiments we chose the following parameters for Algorithm 2: $\tilde{L} = 5000$, $T = 1000$, $\epsilon = 0.005$, and $\gamma_{\text{simul}} = 0.1$ whenever $\gamma = 0$. Generally, we used 50 points to discretize investment levels. To solve $ALP_\theta(\mathcal{R}, \mu, \lambda, c)$, for the capacity model we considered $\theta = 0.32$. We set the parameter to that value, because it provided the best quality of approximations among several values in the comparisons against MPE in the experiments with a small number of firms. We kept that value fixed for the large experiments when comparing against OE. For the quality ladder model, $\theta = 0$ yields good results and so we used that specification for all experiments with that model.

[19] All runs were performed on a shared cluster of (17) computers at Columbia Graduate School of Business. Each node has a 2.4

Parameters	Capacity competition	
	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

Table 2: Parameter selection for comparison with MPE.

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

7.1.2 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, while ignoring 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 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 purpose of our comparisons, we select parameters regimes for which OE provide 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 4 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

Ghz Intel (R) Xeon (R) CPU and 32 Gbs of Ram memory. Our Java implementation called CPLEX 12.1.0 (facilitated by the IBM ILOG Academic Initiative) as a subroutine to solve the linear programs in the algorithms.

Instance		Number of firms		Long-Run Statistics					
				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

Table 3: **Comparison of MPE and ALP-based indicators.** Long-run statistics computed simulating industry evolution over 10^4 periods.

the industry can accommodate). For these instances, computing a MPE using Algorithm 1 is infeasible for $N > 5$.

Parameters	Quality ladder	
	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	

Table 4: Parameter selection for comparison with OE.

Table 5 reports the results for the different parameters 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. While our simulation routine typically sample around $5,000$ states, we consider only the $L = 1500$ most visited states to keep running times relatively low.[20]

We observe that ALP-based indicators are always within 8.5% from 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, while 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.

8 Conclusions and Extensions

The goal of this paper 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

[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. Note that in some of these computations we set $\theta > 0$ so that the effective number of variables in a linear program could be larger than the number of basis functions.

Instance		Number of firms	Long-Run Statistics						
				Total Inv.	Prod. Surp.	Cons. Surp.	C6	C12	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

Table 5: **Comparison of OE and ALP-based indicators.** Long-run statistics computed simulating industry evolution over 10^4 periods.

computed via 'approximate linear 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 up the door to study dynamics in industries for 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, Iskhakov, Rust, and Schjerning (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 the state space that assumes identity of firms do 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 paper, 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 P. Mira (2007). Sequential estimation of dynamic discrete games. *Econometrica 75*(1), 1 – 54.

Applegate, D., R. Bixby, V. Chvatal, and W. Cook (2006). *The traveling Salesman Problem: A Computational study*. Princeton University Press.

Bajari, P., C. L. Benkard, and J. Levin (2007). Estimating dynamic models of imperfect competition. *Econometrica 75*(5), 1331 – 1370.

Bertsekas, D. P. (2001). *Dynamic Programming and Optimal Control, Vol. 2* (Second ed.). Athena Scientific.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Besanko, D. and U. Doraszelski (2004). Capacity dynamics and endogenous asymmetries in firm size. *RAND Journal of Economics 35*(1), 23 – 49.

Calafiore, G. and M. Campi (2005). Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming, Series A 102*(1), 25–46.

Caplin, A. and B. Nalebuff (1991). Aggregation and imperfect competition - on the existence of equilibrium. *Econometrica 59*(1), 25 – 59.

de Farias, D. P. and B. Van Roy (2003). The linear programming approach to approximate dynamic programming. *Operations Research 51*(6), 850 – 865.

de Farias, D. P. and B. Van Roy (2004). On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research 29*(3), 462 – 478.

Desai, V. V., V. F. Farias, and C. C. Moallemi (2010). Aproximate dynamic programming via a smoothed approximate linear program. Working Paper, MIT.

Doraszelski, U. (2003). An r&d race with knowledge accumulation. *RAND Journal of Economics 34*(1), 19 – 41.

Doraszelski, U. and K. Judd (2010). Avoiding the curse of dimensionality in dynamic stochastic games. Working Paper, Harvard University.

Doraszelski, U. and A. Pakes (2007). A framework for applied dynamic analysis in IO. In *Handbook of Industrial Organization, Volume 3*. North-Holland, Amsterdam.

Doraszelski, U. and M. Satterthwaite (2010). Computable markov-perfect industry dynamics. *RAND Journal of Economics 41*(2), 215 – 243.

Ericson, R. and A. Pakes (1995). Markov-perfect industry dynamics: A framework for empirical work. *Review of Economic Studies 62*(1), 53 – 82.

Fudenberg, D. and J. Tirole (1991). *Game Theory*. MIT Press.

Harrington, J. E., F. Iskhakov, J. Rust, and B. Schjerning (2010). A dynamic model of leap-frogging investments and bertrand price competition. Working Paper, University of Maryland.

Judd, K. (1998). *Numerical Methods in Economics*. MIT Press.

Krusell, P. and A. A. Smith, Jr. (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of Political Economy 106*(5), 867 – 896.

Maskin, E. and J. Tirole (1988). A theory of dynamic oligopoly, I and II. *Econometrica 56*(3), 549 – 570.

Pakes, A. and P. McGuire (1994). Computing Markov-perfect Nash equilibria: Numerical implications of a dynamic differentiated product model. *RAND Journal of Economics 25*(4), 555 – 589.

Pakes, A. and P. McGuire (2001). Stochastic algorithms, symmetric Markov perfect equilibrium, and the 'curse' of dimensionality. *Econometrica 69*(5), 1261 – 1281.

Pakes, A., M. Ostrovsky, and S. Berry (2007). Simple estimators for the parameters of discrete dynamic games, with entry/exit examples. *RAND Journal of Economics 38*(2), 373 – 399.

Pesendorfer, M. and P. Schmidt-Dengler (2008). Asymptotic least squares estimators for dynamic games. *Review of Economic Studies 75*(3), 901–928.

Trick, M. and S. Zin (1993). A linear programming approach to solving dynamic programs. Working Paper, Carnegie Mellon University.

Trick, M. and S. Zin (1997). Spline approximations to value functions: A linear programming approach. *Macroeconomic Dynamics 1*.

Tsitsiklis, J. N. and B. Van Roy (1996). Featured-based methods for large scale dynamic programming. *Machine Learning 22*, 59 – 94.

Weintraub, G. Y., C. L. Benkard, and B. Van Roy (2008). Markov perfect industry dynamics with many firms. *Econometrica 76*(6), 1375–1411.

Weintraub, G. Y., C. L. Benkard, and B. Van Roy (2010). Computational methods for oblivious equilibrium. *Operations Research 58*(4), 1247 – 1265.

Whitt, W. (1980). Representation and approximation of noncooperative sequential games. *SIAM J. on Control and Optimization 18*(1), 33–48.

A Details of Specific Models

In this section we provide details regarding the models described in Section 3.2.

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.[21] 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:

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

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,

[21] In our experiments, we eventually consider both settings where $\gamma = 0$ and $\gamma > 0$. We discuss this in more detail in Section 6.1.

summarized in Table 6. We will keep these parameters fixed for all experiments, unless otherwise stated. The values of δ and b are set like in Pakes and McGuire (1994). We also set the mean entry cost to be much higher than the mean sell-off value.

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

Table 6: Default parameters for numerical experiments

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.

A.1 Profit function: Quality Ladder Model

We consider an industry with differentiated products, where each firm's state variable represents the quality of its product. 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:

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

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^* denote such equilibrium; expected profits are given by:

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

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

We use a particular set of representative parameter values, summarized in Table 7, that we keep fixed for all experiments, unless otherwise stated.

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

Table 7: Default parameters for Quality Ladder model.

A.2 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 $q(0) = q_{min}$, and a firm in state $\overline{x}$ has a capacity of $q(\overline{x}) = q_{max}$. 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 $P(Q) = e/f - Q/(mf)$. 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:

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

where $q(x_{it})$ 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^*, which is characterized by the following set of equations:

$$q_{it}^* = \max\left\{0, \min\left\{q(x_{it}), \frac{1}{2}(me - Q_{-i,t}^*)\right\}\right\}, \quad \forall\, i \in s_t.$$

Profits for firm i are then given by: $P(q_{it}^* + Q_{-i,t}^*)q_{it}^*$. We use a particular set of representative parameter values, summarized in Table 8, that we keep fixed for all experiments, unless otherwise stated.

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

Table 8: Default parameters for Capacity model.

B Basis Functions for Separable Approximation Architecture

The family of 'separable' basis functions is easily expressed in the form $\Phi_i : \mathcal{Y} \to \mathbb{R}$, $i = 1, 2, \ldots, K$ where we approximate the value function $V_{\mu,\lambda}^*$ with Φr. For this, we define indicator functions for appropriate sets of states. 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

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

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 $r \in \mathbb{R}^{|\mathcal{X}|^2\cdot(N+1)}$.

C Discretization

We will demonstrate the impact of discretizing sell-off values and investment levels in the context of computing an exact best response to an incumbent strategy (μ,λ). We first consider the impact of discretizing investment levels and then proceed to understand the impact of discretizing sell-off values. For convenience, we will assume that the sell-off value distribution has support $[0,\overline{\kappa}]$ although it is not difficult to extend the analysis here to general continuous distributions.

Let us denote by $V^{*,\epsilon}_{\mu,\lambda}$ the value function corresponding to a best response investment/exit strategy to (μ,λ) when investments are restricted to the set $\mathcal{I}^\epsilon$. With a slight abuse of notation denote this "restricted" best response strategy by $\mu^{*,\epsilon}$; $\mu^{*,\epsilon}$ may be recovered as the greedy strategy with respect to $V^{*,\epsilon}_{\mu,\lambda}$. The value function $V^{*,\epsilon}_{\mu,\lambda}$ is the unique fixed point of the discretized Bellman operator $T^{\epsilon}_{\mu,\lambda}$ defined according to

$$(T^{\epsilon}_{\mu,\lambda}V)(x,s) = \max_{\mu'(x,s):\iota'(x,s)\in\mathcal{I}^\epsilon} (T^{\mu'}_{\mu,\lambda}V)(x,s), \ \forall (x,s)\in\mathcal{Y}. \tag{C.1}$$

and may be computed by the solution of the following program:

$$\begin{aligned} \min \quad & c'V \\ \text{s.t.} \quad & (T^{\epsilon}_{\mu,\lambda}V)(x,s) \le V(x,s) \quad \forall (x,s)\in\mathcal{Y}. \end{aligned} \tag{C.2}$$

The impact of this discretization is given by the following Lemma:

Lemma C.1. *Let* $\tilde{\epsilon} < 1$ *satisfy* $1-\tilde{\epsilon} \le \frac{\mathcal{P}(x_1=x'|x_0=x,\iota_0=\lfloor \iota/\epsilon\rfloor\epsilon)}{\mathcal{P}(x_1=x'|x_0=x,\iota_0=\iota)}$, $\forall x, x', \iota$. *Let* $V^{*,\epsilon}_{\mu,\lambda}$ *be an optimal solution to* (C.2). *Then:*

$$\|V^{*,\epsilon}_{\mu,\lambda} - V^{*}_{\mu,\lambda}\|_\infty \le \frac{\tilde{\epsilon}\beta(\overline{\pi}+\overline{\iota}+\overline{\kappa})}{(1-\beta)^2} + \frac{d\epsilon}{1-\beta}.$$

We next consider our discretization of the sell-off value distribution. By our choice of discretization points, $\hat{\mathcal{K}}$, it is easy to check that

$$\left| \frac{1}{n}\sum_{i=1}^{n}(k_i \vee C) - E[\kappa \vee C] \right| \le \frac{\overline{\kappa}}{n} \ , \ \forall C \in [0, \overline{\pi}/(1-\beta)+\overline{\kappa}], \tag{C.3}$$

Now, suppose firms face a discrete sell-off value distribution and only consider a finite set of investment levels as described before. In this setting, we propose to compute the *exact* best response to (μ, λ) via the program

$$\text{(C.4)} \qquad \begin{array}{ll} \min & c'V \\ \text{s.t.} & (T^{\epsilon,n}_{\mu,\lambda}V)(x,s) \leq V(x,s), \quad \forall (x,s) \in \mathcal{Y}. \end{array}$$

Let us denote by $V^{*,\epsilon,n}_{\mu,\lambda}$ the optimal solution to this linear program. We then have the following Lemma characterizing the approximation to $V^{*,\epsilon}_{\mu,\lambda}$ provided by $V^{*,\epsilon,n}_{\mu,\lambda}$:

Lemma C.2. *We have:*

$$\|V^{*,\epsilon,n}_{\mu,\lambda} - V^{*,\epsilon}_{\mu,\lambda}\|_\infty \leq \frac{\overline{\kappa}}{n(1-\beta)}.$$

The triangle inequality with the proofs of the preceding Lemmas then immediately yield:

$$\|V^{*,\epsilon,n}_{\mu,\lambda} - V^{*}_{\mu,\lambda}\|_\infty \leq \frac{\tilde{\epsilon}\beta(\overline{\pi} + \overline{\iota} + \overline{\kappa})}{(1-\beta)^2} + \frac{d\epsilon}{1-\beta} + \frac{\overline{\kappa}}{n(1-\beta)},$$

which suggests that as the discretization provided by ϵ and n gets sufficiently fine, our approximation to the best response value function to an incumbent policy (μ, λ) gets progressively better.

In the remainder of this section, we will provide proofs of the above Lemmas:

Proof of Lemma C.1. $V^{*,\epsilon}_{\mu,\lambda}$ is the value function corresponding to a best response investment strategy to (μ, λ) when investments in a given time are restricted to the set $\mathcal{I}^\epsilon$. We show that

$$0 \leq V^{*}_{\mu,\lambda} - V^{*,\epsilon}_{\mu,\lambda} \leq \frac{\beta\tilde{\epsilon}(\overline{\pi} + \overline{\iota} + \overline{\kappa})}{(1-\beta)^2} + \frac{d\epsilon}{1-\beta}.$$

Let $P^*_{\mu,\lambda} \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ be a state transition matrix corresponding to using the best response strategy μ^* in response to (μ, λ); note that this is a sub-stochastic matrix since a firm may exit. Define μ^ϵ according to $\iota^\epsilon(x,s) = \lfloor(\iota^*(x,s))/\epsilon\rfloor\epsilon$ and $\rho^\epsilon(x,s) = \rho^*(x,s)$. Let $P^\epsilon_{\mu,\lambda}$ be the corresponding state transition matrix. Moreover, let $g, g^\epsilon \in \mathbb{R}^{|\mathcal{Y}|}$ be respectively defined according to

$$g(x,s) = \pi(x,s) - \mathcal{P}(\kappa < \rho^*(x,s))d\iota^*(x,s) + E[\kappa; \kappa \geq \rho^*(x,s)]$$

and

$$g^\epsilon(x,s) = \pi(x,s) - \mathcal{P}(\kappa < \rho^*(x,s))d\iota^\epsilon(x,s) + E[\kappa; \kappa \geq \rho^*(x,s)].$$

Now since $1 - \tilde{\epsilon} \leq \frac{\mathbb{P}(x_1=x'|x_0=x,\iota_0=\lfloor\iota/\epsilon\rfloor\epsilon)}{\mathbb{P}(x_1=x'|x_0=x,\iota_0=\iota)}$ $\forall x, x', \iota$. by assumption, we must have that

$$P^\epsilon_{\mu,\lambda} = (1-\tilde{\epsilon})P^*_{\mu,\lambda} + \tilde{\epsilon}\hat{P},$$

for some sub-stochastic matrix $\hat{P}$. Given the representation above, we may couple the sample paths under the μ^* and μ^ϵ strategies so that the states visited under both strategies are identical until a random time $\tau^{\tilde{\epsilon}}$ which is distributed as a geometric random variable with mean $1/\tilde{\epsilon}$. Letting $\tilde{V}^*_{\mu,\lambda} = \sum_{t=0}^{\infty} \beta^t \left(P^*_{\mu,\lambda}\right)^t g^\epsilon$, and noting that the maximal absolute difference in the performance of two arbitrary strategies starting from a given state is bounded from above by $\frac{(\overline{\pi}+\overline{\iota}+\overline{\kappa})}{1-\beta}$, this lets us conclude that

$$\|\tilde{V}^*_{\mu,\lambda} - V^{\mu^\epsilon}_{\mu,\lambda}\|_\infty \leq \sum_{t=1}^{\infty} \beta^t \tilde{\epsilon}(1-\tilde{\epsilon})^{t-1} \frac{(\overline{\pi}+\overline{\iota}+\overline{\kappa})}{1-\beta} \leq \frac{\tilde{\epsilon}\beta(\overline{\pi}+\overline{\iota}+\overline{\kappa})}{(1-\beta)^2}.$$

Now by the definition of $\iota^\epsilon(\cdot)$,

$$\|\tilde{V}^*_{\mu,\lambda} - V^*_{\mu,\lambda}\|_\infty \leq \| \sum_{t=0}^{\infty} \beta^t \left(P^*_{\mu,\lambda}\right)^t |g - g^\epsilon| \|_\infty \leq \frac{d\epsilon}{1-\beta}.$$

Thus, by the triangle inequality,

$$\|V^*_{\mu,\lambda} - V^{\mu^\epsilon}_{\mu,\lambda}\|_\infty \leq \frac{\tilde{\epsilon}\beta(\overline{\pi}+\overline{\iota}+\overline{\kappa})}{(1-\beta)^2} + \frac{d\epsilon}{1-\beta}.$$

and since $V^*_{\mu,\lambda} \geq V^{*,\epsilon}_{\mu,\lambda} \geq V^{\mu^\epsilon}_{\mu,\lambda}$, we immediately conclude

$$0 \leq V^*_{\mu,\lambda} - V^{*,\epsilon}_{\mu,\lambda} \leq \frac{\tilde{\epsilon}\beta(\overline{\pi}+\overline{\iota}+\overline{\kappa})}{(1-\beta)^2} + \frac{d\epsilon}{1-\beta}. \tag{C.5}$$

which yields the result. □

Proof of Lemma C.2. First, observe that (C.4) must yield the optimal value of a best response to (μ, λ) when faced with a sell off value distribution that takes values in $\hat{\mathcal{K}}$ uniformly at random; let $\hat{\kappa}$ denote this random variable. It must then be that

$$V^{*,\epsilon,n}_{\mu,\lambda}(x,s) \leq \frac{\overline{\pi}}{1-\beta} + \overline{\kappa} \ , \ \forall \, (x,s) \in \mathcal{Y}.$$

Of course, the same upper bound must hold for $V^{*,\epsilon}_{\mu,\lambda}$. Now, since both κ and $\hat{\kappa}$ are assumed non-negative random variables, taking $\epsilon_n \triangleq \frac{\overline{\kappa}}{n}$, we must then have from (C.3) that

$$\|T^{\epsilon,n}_{\mu,\lambda} V^{*,\epsilon,n}_{\mu,\lambda} - T^{\epsilon}_{\mu,\lambda} V^{*,\epsilon,n}_{\mu,\lambda}\|_\infty \leq \epsilon_n$$

and

$$\|T^{\epsilon,n}_{\mu,\lambda} V^{*,\epsilon}_{\mu,\lambda} - T^{\epsilon}_{\mu,\lambda} V^{*,\epsilon}_{\mu,\lambda}\|_\infty \leq \epsilon_n$$

so that

$$T^{\epsilon}_{\mu,\lambda} V^{*,\epsilon,n}_{\mu,\lambda} \leq V^{*,\epsilon,n}_{\mu,\lambda} + \epsilon_n e$$

and

$$T^{\epsilon,n}_{\mu,\lambda} V^{*,\epsilon}_{\mu,\lambda} \leq V^{*,\epsilon}_{\mu,\lambda} + \epsilon_n e,$$

where e is the vector of all ones. Since $T^{\epsilon}_{\mu,\lambda}(V + \lambda e) \leq T^{\epsilon}_{\mu,\lambda} V + \beta \lambda e$, for $\lambda > 0$, we must then have that

$$\begin{aligned} T^{\epsilon}_{\mu,\lambda}\left(V^{*,\epsilon,n}_{\mu,\lambda} + \frac{\epsilon_n}{1-\beta} e\right) &\leq T^{\epsilon}_{\mu,\lambda} V^{*,\epsilon,n}_{\mu,\lambda} + \frac{\beta \epsilon_n}{1-\beta} e \\ &\leq V^{*,\epsilon,n}_{\mu,\lambda} + \epsilon_n e + \frac{\beta \epsilon_n}{1-\beta} e \\ &= V^{*,\epsilon,n}_{\mu,\lambda} + \frac{\epsilon_n}{1-\beta} e. \end{aligned}$$

Since $T^{\epsilon}_{\mu,\lambda} V \leq V \implies V^{*,\epsilon}_{\mu,\lambda} \leq V$ (by iterating the Bellman operator), it follows that

$$V^{*,\epsilon}_{\mu,\lambda} \leq V^{*,\epsilon,n}_{\mu,\lambda} + \frac{\epsilon_n}{1-\beta} e.$$

Similarly, using the fact that $T^{\epsilon,n}_{\mu,\lambda}(V + \lambda e) \leq T^{\epsilon,n}_{\mu,\lambda} V + \beta \lambda e$, for $\lambda > 0$, we may show that

$$V^{*,\epsilon,n}_{\mu,\lambda} \leq V^{*,\epsilon}_{\mu,\lambda} + \frac{\epsilon_n}{1-\beta} e.$$

The result follows. □

D Approximate Dynamic Programming Theory Background

D.1 Value Function Approximation

The program (5.2) seeks to approximate the value function corresponding to a best response to (μ, λ) within the linear span of a small number of basis functions. In particular, we sought the approximation $\Phi r \sim V^*_{\mu,\lambda}$. The following Theorem (Theorem 2 of de Farias and Van Roy (2003)) demonstrates the sense in which (5.2) actually accomplishes this approximation:

Theorem D.1. *Let $e \in \mathbb{R}^{|\mathcal{Y}|}$, the vector of ones, be in the span of the columns of Φ and c be a probability distribution. Let $r_{\mu,\lambda}$ be an optimal solution to* (5.2). *Then,*[22]

$$\|\Phi r_{\mu,\lambda} - V^*_{\mu,\lambda}\|_{1,c} \leq \frac{2}{1-\beta} \inf_r \|\Phi r - V^*_{\mu,\lambda}\|_\infty.$$

The above theorem shows that as the basis function architecture Φ grows 'richer' in the sense that its span contains a good approximation to $V^*_{\mu,\lambda}$, the approximation computed by the program (5.2) also provides a good approximation to the optimal value function. In fact, the latter is of a quality comparable to the best possible approximation to $V^*_{\mu,\lambda}$ within the span of the basis functions.

[22] For $c \in \mathbb{R}^k_+$, the $(1, c)$ norm of a vector $x \in \mathbb{R}^k$ is defined according to $\|x\|_{1,c} = \sum_{i=1}^k |x_i| c_i$.

D.2 Approximation Error and State Relevance Weights

Given a good approximation to $V^*_{\mu,\lambda}$, namely $\Phi r_{\mu,\lambda}$ 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 is also close to $V^*_{\mu,\lambda}$ as is made precise by the following result which is easy to establish and whose proof is omitted (see de Farias and Van Roy (2003)). Let us denote by $P_{\mu';(\mu,\lambda)}$ a transition matrix over the state space $\mathcal{Y}$ induced by using investment/exit strategy μ' in response to (μ,λ). Notice that the matrix $P_{\mu';(\mu,\lambda)}$ is sub-stochastic since the firm may exit. Now, denote by $c^{\mu'}_{\mu,\lambda}$ the sub-probability distribution

$$(1-\beta)\sum_{t=0}^{\infty}\beta^t\nu^\top P^t_{\mu';(\mu,\lambda)}. \tag{D.1}$$

$c^{\mu'}_{\mu,\lambda}$ describes the discounted relative frequency with which states in $\mathcal{Y}$ are visited during a firms lifetime in the industry assuming a starting state over $\mathcal{Y}$ distributed according to ν.

Theorem D.2. *Given strategies (μ,λ), and defining $\tilde{\mu}$ and $c^{\tilde{\mu}}_{\mu,\lambda}$ as above for an arbitrary distribution over initial states in $\mathcal{Y}$, ν , we have:*

$$\|V^{\tilde{\mu}}_{\mu,\lambda} - V^*_{\mu,\lambda}\|_{1,\nu} \leq \frac{1}{1-\beta}\|\Phi r_{\mu,\lambda} - V^*_{\mu,\lambda}\|_{1,c^{\tilde{\mu}}_{\mu,\lambda}} .$$

Together with Theorem D.1, this result lets us conclude that

$$\|V^{\tilde{\mu}}_{\mu,\lambda} - V^*_{\mu,\lambda}\|_{1,\nu} \leq \max_{(x,s)\in\mathcal{Y}} \frac{c^{\tilde{\mu}}_{\mu,\lambda}(x,s)}{c(x,s)}\frac{2}{(1-\beta)^2}\inf_r \|\Phi r - V^*_{\mu,\lambda}\|_\infty. \tag{D.2}$$

Let us discuss what we have established. First, the previous expression provides a bound on how much a firm can improve its expected discounted profits by unilaterally deviating from the strategy derived by the approximate dynamic programming approach to an optimal strategy. In particular, upon convergence of our approximate best response algorithm, the extent of this unilateral deviation will be small if the chosen basis functions provide an accurate approximation to the optimal value function when competitors use the candidate equilibrium strategy. This measure of the accuracy of the approximation is related to the notion of an $\epsilon-$equilibrium. We formalize this notion in Section E and relate it to the convergence to MPE strategies.

Second, the state relevance weight vector c plays the role of trading off approximation error across states which follows from the fact that (5.2) is equivalent to the program (see de Farias and Van Roy (2003)):

$$\begin{aligned} \min \quad & \|\Phi r - V^*_{\mu,\lambda}\|_{1,c} \\ \text{s.t.} \quad & (T_{\mu,\lambda}\Phi r)(x,s) \leq (\Phi r)(x,s) \quad \forall (x,s) \in \mathcal{Y}. \end{aligned}$$

In fact, expression (D.2) suggests that the vector c should ideally assign weights to industry states according to $c^{\tilde{\mu}}_{\mu,\lambda}$. To be more concrete, suppose one is interested in approximating the long-run behavior of the industry in the sense that $V^{\tilde{\mu}}_{\mu,\lambda}$ and $V^*_{\mu,\lambda}$ are close when weighting industry states according to the invariant distribution of the Markov process that describes the industry evolution under strategies (μ, λ). Then, the selection of ν in $c^{\tilde{\mu}}_{\mu,\lambda}$ (and hence in c) should weight industry states according to the invariant distribution. Note that in this case c itself will weight industry states according to the invariant distribution.[23] 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.

D.3 Reducing the Number of Constraints

Here we describe an 'idealized' sampling distribution ψ^* under which the program (5.3) provides a good approximation to $V^*_{\mu,\lambda}$. In particular, assume we had access to the strategy $\mu^*_{\mu,\lambda}$ and consider sampling states according to a distribution ψ^* defined according to $\psi^*(x,s) = c^{\mu^*}_{\mu,\lambda}(x,s) / \sum_{(x',s') \in \mathcal{Y}} c^{\mu^*}_{\mu,\lambda}(x',s')$ where we take ν to be equal to the state relevant weights vector, c, in the definition of $c^{\mu^*}_{\mu,\lambda}$ (see equation (D.1)). We do not have access to ψ^*; let $\overline{\psi}$ be a sampling distribution satisfying $\max_{x,s} \frac{\psi^*(x,s)}{\overline{\psi}(x,s)} \leq M$. Assuming the L states in $\mathcal{R}$ are sampled according to $\overline{\psi}$, we then have the following result.

Theorem D.3. *Let δ, ϵ' be arbitrary numbers in $(0,1)$. Let $\mathcal{R}$ consist of L states in $\mathcal{Y}$ sampled according to $\overline{\psi}$. Let $\tilde{r}_{\mu,\lambda}$ be an optimal solution to* (5.3)*. If*

$$L \geq \frac{KM}{\epsilon'\delta} \frac{2(1+\beta)}{c^\top V^*_{\mu,\lambda}(1-\beta)} \|\Phi\tilde{r}_{\mu,\lambda} - V^*_{\mu,\lambda}\|_\infty$$

then, with probability at least $1-\delta$, we have

$$\|V^*_{\mu,\lambda} - \Phi\tilde{r}_{\mu,\lambda}\|_{1,c} \leq \|V^*_{\mu,\lambda} - \Phi r_{\mu,\lambda}\|_{1,c} + \epsilon'\|V^*_{\mu,\lambda}\|_{1,c},$$

where $r_{\mu,\lambda}$ solves (5.2)*.*

The proof of the above result closely follows that of Theorem 3.1 in de Farias and Van Roy (2004); since the program (5.3) is distinct from that studied by de Farias and Van Roy (2004), their result does not apply directly. In particular, since (5.3) is a convex program (but not an LP), we employ the sample complexity

[23] Under the assumptions in our model such an invariant distribution always exists and is unique.

bound (Theorem 3) of Calafiore and Campi (2005) in lieu of Theorem 2.1 of de Farias and Van Roy (2004); the proof is then essentially identical to that of their Theorem 3.1.[24] The result and the discussion in de Farias and Van Roy (2004) suggest that sampling a tractable number of constraints according to a distribution close to ψ^* ensures that $\|V^*_{\mu,\lambda} - \Phi\tilde{r}_{\mu,\lambda}\|_{1,c} \approx \|V^*_{\mu,\lambda} - \Phi r_{\mu,\lambda}\|_{1,c}$.[25] Of course, we do not have access to ψ^*; ψ^* requires we already have access to a best response to (μ, λ). Nonetheless, our sequential MPE computation yields a natural candidate for $\overline{\psi}$: in every iteration we simply sample industry states according to $c^{\mu}_{\mu,\lambda}$ where (μ, λ) are the approximate best response strategy computed at the prior iteration.

D.4 A Heuristic LP Solver for (5.3)

Proposition 5.1. *(r', u', t', l') is an optimal solution to the LP* (6.1)*.*

Proof. Let (r, u, t, l) be a feasible solution to (6.1). It is easy to see that (r, l) is a feasible solution to the LP solved in step (5) of Algorithm 3, (6.2) of the same value, for *any* e_j. Consequently the value of an optimal solution (r^*, u^*, t^*, l^*) to (6.1) is no larger than the value of an optimal solution to (6.2) for any e_j.

Now, (r', u', t', l') is by construction a feasible solution to (6.1). To see this simply note that (6.1) is equivalent to the program

$$
\begin{array}{ll}
\underset{r,l,e}{\text{minimize}} & \displaystyle\sum_{(x,s)\in\mathcal{R}} c(x,s) \sum_{0\le k\le K} \Phi_k(x,s)\, r_k \\
\text{subject to} & \displaystyle \pi(x,s) + \mathcal{P}(\hat{\kappa} < e_j(x,s)) \left(-d\iota + \beta E_{\mu,\lambda}\left[\sum_{0\le k\le K} \Phi_k(x_1,s_1)\, r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right]\right) \\
& \qquad \displaystyle +E[\hat{\kappa}|\hat{\kappa} \ge e_j(x,s)]\mathcal{P}(\hat{\kappa} \ge e_j(x,s)) \le \sum_{0\le k\le K} \Phi_k(x,s)\, r_k + l(x,s)\ , \\
& \qquad \forall (x,s) \in \mathcal{R}, \iota \in \mathcal{I} \\
& e_j(x,s) = \max_{\iota\in\mathcal{I}} -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0\le k\le K} \Phi_k(x_1,s_1)\, r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right], \\
& \qquad \forall (x,s) \in \mathcal{R}, \\
& \frac{1}{|\mathcal{R}|} \sum_{(x,s)\in\mathcal{R}} l(x,s) \le \theta \\
& l(x,s) \ge 0 \quad \forall (x,s) \in \mathcal{R}.
\end{array}
$$

and that upon convergence in Algorithm 3,

$$
e_j(x,s) = e_{j+1}(x,s) = \max_{\iota\in\mathcal{I}} -d\iota + \beta E_{\mu,\lambda}\left[\sum_{0\le k\le K} \Phi_k(x_1,s_1)\, r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right].
$$

for all $(x, s) \in \mathcal{R}$. Moreover, this feasible solution to (6.1) has the same value as an optimal solution to (6.2) with $e_j = t'$. Since the value of a feasible solution to (6.1) cannot exceed the value of an optimal solution to

[24] While we have not discussed these technicalities here, Theorem 3 of Calafiore and Campi (2005) requires a certain 'tie-breaking' rule in the event that (5.3) has multiple optimal solutions.

[25] The number grows linearly in the number of basis functions and is independent from the total number of constraints.

(6.2) for any value of e_j, it follows that (r', u', t', l') must be an optimal solution to (6.1) which proves the result. □

E ϵ-Weighted MPE and Approximating MPE

Motivated by the discussion at the end of Section D.2, we begin by defining a notion of ϵ-weighted MPE. Let ν be a given distributions over $\mathcal{Y}$ and let $\hat{\nu}$ be distribution induced by ν on the set $\mathcal{S}^e = \left\{ s \in \mathbb{N}^{|\mathcal{X}|} : \sum_{x=0}^{\overline{x}} s(x) < N \right\}$. Notice that $\mathcal{S}^e$ is the set of industry states that have potential entrants. Given these distributions, we define approximate notions of a best response and an MPE:

Definition E.1. *For $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, we call $(\tilde{\mu}, \tilde{\lambda}) \in \mathcal{M} \times \Lambda$ an $\epsilon-$weighted best response to (μ, λ) if*

$$\text{(E.1)} \qquad \|V^*_{\mu,\lambda} - V^{\tilde{\mu}}_{\mu,\lambda}\|_{1,\nu} \leq \epsilon,$$

$$\text{(E.2)} \qquad \left\| \tilde{\lambda} - \beta E_{\mu,\lambda} \left[V^{\tilde{\mu}}_{\mu,\lambda}(x^e, s_{t+1}) | s_t = \cdot \right] \right\|_{1,\hat{\nu}} \leq \epsilon.$$

Definition E.2. $(\tilde{\mu}, \tilde{\lambda}) \in \mathcal{M} \times \Lambda$ *is an $\epsilon-$weighted MPE if $(\tilde{\mu}, \tilde{\lambda})$ is an $\epsilon-$weighted best response to itself.*

Under the definition above, the maximum potential gain to an incumbent firm in deviating from an $\epsilon-$weighted MPE strategy, $\tilde{\mu}$, is averaged across industry states under the measure ν. Similarly, the potential entrants' strategy, $\tilde{\lambda}$, is such that the zero expected discounted profits entry condition is not satisfied exactly; however, the average error under the measure $\hat{\nu}$ is at most ϵ. This notion of ϵ-weighted MPE is similar to other concepts that have been previously used to assess the accuracy of approximations to MPE (Weintraub, Benkard, and Van Roy 2008) and as stopping criteria (Pakes and McGuire 2001).

It is only natural to ask whether our definition of $\epsilon-$weighted MPE is computationally relevant. Here we note that with an appropriate selection of stopping criterion in our iterative algorithm for equilibrium computation, one may conclude that upon termination we have arrived at an ϵ-weighted MPE where ϵ can be controlled. In particular, consider stopping the algorithm when $\|V^{\mu_{i+1}}_{\mu_i,\lambda_i} - V^{\mu_i}_{\mu_i,\lambda_i}\|_{1,\nu} \leq \epsilon_1$ and $\|\lambda_i - \lambda_{i+1}\|_{1,\hat{\nu}} + \|\beta E_{\mu_i,\lambda_i} \left[V^{\mu_{i+1}}_{\mu_i,\lambda_i}(x^e, s_{t+1}) | s_t = \cdot \right] - \beta E_{\mu_i,\lambda_i} \left[V_{\mu_i,\lambda_i}(x^e, s_{t+1}) | s_t = \cdot \right] \|_{1,\hat{\nu}} \leq \epsilon_1$, where ϵ_1 is some pre-specified tolerance (in our algorithm we use a relaxation of this criteria that worked well in practice; see second comment in Section 6.1). Moreover, let us assume that $\|V^*_{\mu_i,\lambda_i} - V^{\mu_{i+1}}_{\mu_i,\lambda_i}\|_{1,\nu} \leq \epsilon_2$. Notice that our theoretical development of approximate dynamic programming in the previous appendix attempted to characterize precisely the conditions under which ϵ_2 was small; in particular, see (D.2). It is then simple to show using the triangle inequality that we are then guaranteed that upon termination, our algorithm would have computed an $(\epsilon_1 + \epsilon_2)$-MPE.

Now, in what sense does an ϵ-weighted MPE as defined above approximate MPE? We attempt to make this precise under the assumption that ν places positive mass over all states in its support. In what follows, let

$\Gamma \subseteq \mathcal{M} \times \Lambda$ be the set of MPE. For all $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, let us define $\mathcal{D}(\Gamma, (\mu, \lambda)) = \inf_{(\mu', \lambda') \in \Gamma} \|(\mu', \lambda') - (\mu, \lambda)\|_\infty$. We have the following theorem:

Theorem E.1. *Suppose $\nu > 0$. Given a sequence of real numbers $\{\epsilon_n \geq 0 | n \in \mathbb{N}\}$, let $\{(\mu_n, \lambda_n) \in \mathcal{M} \times \Lambda | n \in \mathbb{N}\}$ be a sequence of ϵ_n-weighted MPE with $\lim_{n \to \infty} \epsilon_n = 0$. Then, $\lim_{n \to \infty} \mathcal{D}(\Gamma, (\mu_n, \lambda_n)) = 0$.*

Proof. Assume the claim to be false. It must be that there exists an $\epsilon > 0$ such that for all n, there exists an $n' > n$ for which $d(\Gamma, (\mu_{n'}, \lambda_{n'})) > \epsilon$. We may thus construct a subsequence $\{(\tilde{\mu}_n, \tilde{\lambda}_n)\}$ for which $\inf_n d(\Gamma, (\tilde{\mu}_n, \tilde{\lambda}_n)) > \epsilon$. Now, since the space of strategies is compact, we have that $\{(\tilde{\mu}_n, \tilde{\lambda}_n)\}$ has a convergent subsequence; call this subsequence $\{(\mu'_n, \lambda'_n)\}$ and its limit (μ^*, λ^*).[26] We have thus established the existence of a sequence of strategies and entry rate functions $\{(\mu'_n, \lambda'_n)\}$, satisfying:

$$\|V^{\mathcal{BR}(\mu'_n, \lambda'_n)}_{\mu'_n, \lambda'_n} - V^{\mu'_n}_{\mu'_n, \lambda'_n}\|_{1,\nu} \to 0, \tag{E.3}$$

$$\|\lambda'_n - \beta E_{\mu'_n, \lambda'_n}\left[V^{\mu'_n}_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = \cdot\right]\|_{1,\hat{\nu}} \to 0, \tag{E.4}$$

$$(\mu'_n, \lambda'_n) \to (\mu^*, \lambda^*). \tag{E.5}$$

$$\inf_n d(\Gamma, (\mu'_n, \lambda'_n)) > \epsilon, \tag{E.6}$$

where $\mathcal{BR}(\mu, \lambda)$ denotes the best response strategy when competitors play strategy μ and enter according to λ. Now, since by assumption, we must have $\nu, \hat{\nu} > 0$ component-wise, this implies that

$$V^{\mathcal{BR}(\mu'_n, \lambda'_n)}_{\mu'_n, \lambda'_n}(x, s) - V^{\mu'_n}_{\mu'_n, \lambda'_n}(x, s) \quad \to \quad 0, \ \forall (x, s) \in \mathcal{Y} \tag{E.7}$$

$$\lambda'_n(s) - \beta E_{\mu'_n, \lambda'_n}\left[V^{\mu'_n}_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = s\right] \quad \to \quad 0, \ \forall s \in \mathcal{S}^e. \tag{E.8}$$

Now, it is simple to show that our assumptions on model primitives guarantee that $V^{\mu'}_{\mu, \lambda}(x, s)$ is continuous in (μ', μ, λ) for all $(x, s) \in \mathcal{Y}$, so that $V^{\mu}_{\mu, \lambda}$ is continuous in (μ, λ). Moreover, the assumption of unique investment choice admissibility yields in addition that $\mathcal{BR}(\cdot)$ is continuous on $\mathcal{M} \times \Lambda$. For a proof of this fact, see the proof of Proposition 2 in (Doraszelski and Satterthwaite 2010) which in turn employs Lemmas 3.1 and 3.2 of (Whitt 1980). Thus, we have from (E.7), (E.8), and (E.5) that $V^{\mathcal{BR}(\mu^*, \lambda^*)}_{\mu^*, \lambda^*}(x, s) - V_{\mu^*, \lambda^*}(x, s) = 0, \ \forall (x, s) \in \mathcal{Y}$, and $\lambda^*(s) - \beta E_{\mu^*, \lambda^*}\left[V_{\mu^*, \lambda^*}(x^e, s_{t+1}) | s_t = s\right] = 0, \ \forall s \in \mathcal{S}^e$. Hence, $(\mu^*, \lambda^*) \in \Gamma$. But by (E.6), (E.5) and the triangle inequality $d(\Gamma, (\mu^*, \lambda^*)) > \epsilon$, a contradiction. The result follows. □

[26] Under our assumptions on model primitives, without loss of generality we can restrict attention to bounded exit and entry cutoff strategies. Indeed, it is easily shown that $|\rho(x, s)|$ and $|\lambda(s)|$ are uniformly bounded by $(\overline{\pi} + \hat{\kappa})/(1 - \beta)$. Here $\hat{\kappa} = E[\kappa | \kappa \geq \overline{\pi}/(1 - \beta)]$ is an upper bound on the expected scrap value upon a firm's exit.

We note that the assumption $\nu > 0$ is key to show the result, because it guarantees the extent of a unilateral deviation becomes small in all states as ϵ_n converges to zero. One notes, however, that in the course of our algorithm the weight vector ν used to evaluate the stopping criterion alluded to above changes at each iteration; the relevant distribution over states might be assumed, for instance, to be the long run distribution under the incumbent policy in that iteration. The result we have just established can be extended to this case. The extension, however, requires the additional assumption that for all strategies $\mu \in \mathcal{M}$ and entry rate functions $\lambda \in \Lambda$, the Markov chain that describes the industry state evolution $\{s_t : t \geq 0\}$ is irreducible and aperiodic. Together with the fact that the state space is finite, it implies that the Markov chain $\{s_t : t \geq 0\}$ admits a unique invariant distribution that assigns strictly positive mass to all states.[27]

[27] The assumption is satisfied if, for example, (i) for all strategies and all states $(x, s) \in \mathcal{Y}$, there is a strictly positive probability that an incumbent firm will visit state x at least once before exiting, and $\pi(x, s) \geq 0$; and (ii) exit and entry cut-off values are restricted to belong to the sets $[0, \max_{x,s} \pi(x, s)/(1 - \beta) + \hat{\kappa}]$ and $[\hat{\kappa}, \infty)$, respectively, where $\hat{\kappa}$ is the expected net present value of entering the market, investing zero and earning zero profits each period, and then exiting at an optimal stopping time. Note that the latter assumption is not very restrictive as all best response exit/entry strategies lie in that set. To satisfy the former assumption, the model needs to have depreciation and appreciation shocks in all states, as we assume in some of our numerical experiments.