

An Approximate Dynamic Programming Approach to Solving Dynamic Oligopoly Models*

Vivek Farias
MIT Sloan School

Denis Saure
Columbia Business School

Gabriel Y. Weintraub
Columbia Business School

January, 2010

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 based on linear programming. We provide results that lend theoretical support to our approximation. We test our method on an important class of models based on Pakes and McGuire (1994). 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 (2003), Bajari, Benkard, and Levin (2007), Pakes, Ostrovsky, and Berry (2007), Aguirregabiria and Mira (2007)) this has also become an active area for empirical research.

*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, 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. The research of the first author was supported, in part, by the Solomon Buchsbaum Research Fund. Correspondence: vivekf@mit.edu, dsaure05@gsb.columbia.edu, gweintraub@columbia.edu

¹Indeed, 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).

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 computations 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 (2006)). 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.

Our alternative approach is based on an algorithm that iterates an ‘approximate’ best response operator. 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)). In short, the value function is approximated by a linear combination of basis functions. In each iteration, the weights

associated to these basis functions are computed by solving a suitable linear program; these weights yield the approximate best response value function. Our scheme iteratively computes approximations to the best response via a *tractable* algorithm 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 a class of EP-style models. We next outline our contributions in detail.

Our first main contribution is to provide a tractable algorithm to approximate MPE in large scale EP-style dynamic oligopoly models. We present an easy to follow guide to using the algorithm in EP-style models. Among other things, we carefully address several implementation issues, such as, constraint sampling for the linear programming approach and strategy storage between iterations. Our algorithm runs in few hours on a modern workstation,² even in models with tens of firms and tens of quality levels per firm. This presentation should appeal to practitioners of the approach.

We provide an extensive computational demonstration of our method where we attempt to show that it works well in practice on a class of EP-style models motivated by Pakes and McGuire (1994). A similar model has been previously used as a test bed for new methods to compute and approximate MPE (Doraszelski and Judd (2006), and Weintraub, Benkard, and Van Roy (2009)). 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. 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 approximation architecture and a suitable version of our approximate best response algorithm, we explore the problem of approximating MPE across various problem regimes.

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 quality levels 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

²The workstation used had an Intel(R) Xeon(R) (X5365 3.00GHz) processor and 32GB of RAM.

equilibrium strategy is close to OE and hence to MPE.

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. Moreover, with the theoretical performance guarantees we present for our approach, this in turn suggests that upon convergence our scheme 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.

Our second main contribution 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.

This result is the synthesis of several results as we now explain. An extension of the theory developed in de Farias and Van Roy (2003) and de Farias and Van Roy (2004) lets us bound the magnitude by which a firm can increase its expected discounted payoffs, by unilaterally deviating from a strategy produced by the approach to a best response strategy, in terms of the expressivity of the approximation architecture. It is worth noting that such bounds 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.

Bounds of the style described above are related to the notion of an ϵ -equilibrium (Fudenberg and Tirole 1991) and provide a useful metric to assess the accuracy of the approximation. Under an additional assumption, we demonstrate a relationship between the notion of ϵ -equilibrium and approximating equilibrium strategies that provides a more direct test of the accuracy of our approximation. In Theorem 3.1 we show that if the Markov chain that describes the industry evolution is irreducible under any strategy, then 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.

As we have discussed above, our work is related to Weintraub, Benkard, and Van Roy (2008) and Weintraub, Benkard, and Van Roy (2009). 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. Perakis, Kachani, and Simon (2008) explore the use of linear and quadratic approximations to the value function in a duopoly dynamic pricing game. 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 high 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.

The paper is organized as follows. In Section 2 we introduce our dynamic oligopoly model. In Section 3 we discuss computation and approximation of MPE. In Section 4 we describe our approximate linear programming approach and provide approximation bounds. In Section 5 we provide a ‘guide for practitioners’ of our algorithm. In Section 6 we report results from computational experiments. In Section 7 we provide conclusions and discuss extensions of our work.

2 A Dynamic Oligopoly Model

In this section we formulate a model of an industry in which firms compete in a single-good market. The model closely follows Weintraub, Benkard, and Van Roy (2008) which in turn, is close in spirit to Ericson and Pakes (1995). The entry process and the structure of industry-wide shocks is similar to Doraszelski and Pakes (2007).

2.1 Model and Notation

The industry evolves over discrete time periods and an infinite horizon. We index time periods with non-negative integers $t \in \mathbb{N}$ ($\mathbb{N} = \{0, 1, 2, \dots\}$). All random variables are defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ equipped with a filtration $\{\mathcal{F}_t : t \geq 0\}$. We adopt a convention of indexing by t variables that are \mathcal{F}_t -measurable.

Each incumbent firm is assigned a unique positive integer-valued index. The set of indices of incumbent firms at time t is denoted by S_t . Firm heterogeneity is reflected through firm states. To fix an interpretation, we will refer to a firm's state as its quality level. However, firm states might more generally reflect productivity, capacity, the size of its consumer network, or any other aspect of the firm that affects its profits. At time t , the quality level of firm $i \in S_t$ is denoted by $x_{it} \in \mathcal{X} = \{0, 1, 2, \dots, \bar{x}\}$. The integer number \bar{x} is an upper bound on firms' quality levels.

We define the *industry state* s_t to be a vector over quality levels that specifies, for each quality level $x \in \mathcal{X}$, the number of incumbent firms at quality level x in period t . We define the state space $\mathcal{S} = \left\{s \in \mathbb{N}^{|\mathcal{X}|} \mid \sum_{x=0}^{\bar{x}} s(x) \leq N\right\}$. 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, $n_t = \sum_{x=0}^{\bar{x}} s_t(x)$.

In each period, each incumbent firm earns profits on a spot market. For firm $i \in S_t$, its single period expected profits $\pi(x_{it}, s_t)$ depend on its quality level $x_{it} \in \mathcal{X}$ and the industry state $s_t \in \mathcal{S}$.

The model also allows for entry and exit. In each period, each incumbent firm $i \in S_t$ 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.

If the firm instead decides to remain in the industry, then it can invest to improve its quality level. If a

firm invests $l_{it} \in \mathbb{R}_+$, then the firm's state at time $t + 1$ is given by,

$$x_{i,t+1} = \min(\bar{x}, \max(0, x_{it} + w(l_{it}, \zeta_{i,t+1}) + \eta_{t+1})),$$

where the function w captures the impact of investment on quality and $\zeta_{i,t+1}$ reflects idiosyncratic uncertainty in the outcome of investment. Uncertainty may arise, for example, due to the risk associated with a research and development endeavor or a marketing campaign. The random variable η_{t+1} represents industry-wide shocks that, for example, change the desirability of the outside alternative. The latter shocks are exogenous and equally affect all goods produced in the industry. Note that this specification is very general as w and η_t may take on either positive or negative values (e.g., allowing for positive depreciation). We denote the unit cost of investment by d .

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. The set of indices of potential entrants at time t is denoted by S'_t . In each time period each potential entrant $i \in S'_t$ observes a positive real-valued entry cost ϕ_{it} that is private information to the firm. 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.

Each firm aims 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.

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.
5. Investment and industry-wide shock outcomes are determined, new entrants enter, and the industry takes on a new state s_{t+1} .

³We assume $n_0 \leq N$.

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

The model is general enough to encompass numerous applied problems in economics.⁵ To study any particular problem it is necessary to further specify the primitives of the model, including the profit function π , the sell-off value distribution κ_{it} , the investment impact function w , the investment uncertainty distribution ζ_{it} , the industry-wide shock distribution η_t , the unit investment cost d , the entry cost distribution ϕ_{it} , and the discount factor β . 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.

2.2 Assumptions

We make several assumptions about the model primitives.

Assumption 2.1.

1. *There exists $\bar{\pi} < \infty$, such that, $|\pi(x, s)| < \bar{\pi}$, for all $x \in \mathcal{X}$, $s \in \mathcal{S}$.*
2. *The variables $\{\kappa_{it} | t \geq 0, i \geq 1\}$ are i.i.d. and have a well-defined density function with support $[0, \bar{\kappa}]$, for some $\bar{\kappa} > 0$.*
3. *The random variables $\{\zeta_{it} | t \geq 0, i \geq 1\}$ are i.i.d. and independent of $\{\kappa_{it} | t \geq 0, i \geq 1\}$.*
4. *The random variables $\{\eta_t | t \geq 0\}$ are i.i.d. and independent of $\{\kappa_{it}, \zeta_{it} | t \geq 0, i \geq 1\}$.*
5. *The random variables $\{\phi_{it} | t \geq 0, i \geq 1\}$ are i.i.d. and independent of $\{\kappa_{it}, \zeta_{it}, \eta_t | t \geq 0, i \geq 1\}$. They have a well-defined density function with support $[0, \bar{\phi}]$, for some $\bar{\phi} > 0$.*
6. *For all ζ , $w(\iota, \zeta)$ is nondecreasing in ι . For all $\iota > 0$, $\mathcal{P}[w(\iota, \zeta_{i,t+1}) > 0] > 0$.*
7. *There exists a positive constant $\bar{\iota}$ such that $\iota_{it} \leq \bar{\iota}$, $\forall i, \forall t$.*
8. *For all k , $\mathcal{P}[w(\iota, \zeta_{i,t+1}) = k]$ is continuous in ι .*
9. *The transitions generated by $w(\iota, \zeta)$ are unique investment choice admissible.*

The assumptions are natural and fairly weak. Assumption 2.1.1 ensures profits are bounded. Assumptions 2.1.2, 2.1.3, 2.1.4, and 2.1.5 impose some probabilistic structure over the sell-off value, entry cost, and the idiosyncratic and aggregate shocks. Assumption 2.1.6 implies that investment is productive. Assumption 2.1.7 places a bound on investment levels. Assumption 2.1.8 ensures that the impact of investment on transition probabilities is continuous. Assumption 2.1.9 is an assumption introduced by Doraszelski and Satterthwaite (2007) that 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

⁵Indeed, a blossoming recent literature on EP-style models has applied similar models to advertising, auctions, collusion, consumer learning, environmental policy, international trade policy, learning-by-doing, limit order markets, mergers, network externalities, and other applied problems (see Doraszelski and Pakes (2007)).

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.

Assumption 2.1 is kept throughout the paper unless otherwise explicitly noted.

2.3 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 $i \in S_t$ 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 $i \in S_t$ exits at time t if and only if $\kappa_{it} \geq \rho(x_{it}, s_t)$. 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 $\mathcal{Y} = \{(x, s) \in \mathcal{X} \times \mathcal{S} : s(x) > 0\}$. Let \mathcal{M} denote the set of exit/investment strategies such that an element $\mu \in \mathcal{M}$ is a set of functions $\mu = (\iota, \rho)$, where $\iota : \mathcal{Y} \rightarrow \mathbb{R}_+$ is an investment strategy and $\rho : \mathcal{Y} \rightarrow \mathbb{R}$ is an exit strategy.

Similarly, each potential entrant follows an entry strategy that takes the form of a cutoff rule: there is a real-valued function λ such that a potential entrant $i \in S'_t$ 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 (2007)). We denote the set of entry strategies by Λ , where an element of Λ is a function $\lambda : \mathcal{S}^e \rightarrow \mathbb{R}$ and $\mathcal{S}^e = \{s \in \mathcal{S} : \sum_{x=0}^{\bar{x}} s(x) < N\}$.

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

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

where i is taken to be the index of a firm at quality level x at time t , τ_i is a random variable representing the time at which firm i exits the industry, and the superscript and subscripts of the expectation indicate the strategy followed by firm i , the strategy followed by its competitors, and the entry strategy. In an abuse of notation, we will use the shorthand, $V_{\mu, \lambda}(x, s) \equiv V_{\mu, \lambda}^{\mu}(x, s)$, to refer to the expected discounted value of profits when firm i follows the same strategy μ as its competitors.

An equilibrium to 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:

$$(2.1) \quad \sup_{\mu' \in \mathcal{M}} V_{\mu, \lambda}^{\mu'}(x, s) = V_{\mu, \lambda}(x, s), \quad \forall (x, s) \in \mathcal{Y}.$$

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:⁶

$$(2.2) \quad \lambda(s) = \beta E_{\mu, \lambda} [V_{\mu, \lambda}(x^e, s_{t+1}) | s_t = s], \quad \forall s \in \mathcal{S}^e.$$

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 (2007) 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.⁷

3 Computing and Approximating MPE

In this section we introduce a best response algorithm to compute MPE. 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. We provide a theoretical justification for our approach by relating the approximate best responses to the notion of ϵ -weighted MPE that we introduce below. Moreover, we show in Theorem 3.1 that under some additional assumptions, as we improve the accuracy of the approximate best responses, we get closer to a 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 $\sup_{\mu' \in \mathcal{M}} V_{\mu, \lambda}^{\mu'} = V_{\mu, \lambda}^{\mu_{\mu, \lambda}^*}$, where the supremum is attained point-wise. With some abuse of notation we will usually denote the best response to (μ, λ) by simply μ^* . We also define the best response

⁶Hence, 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}) .

⁷Doraszelski and Satterthwaite (2007) also provide an example of multiple equilibria in a closely related model.

value function as $V_{\mu,\lambda}^* = V_{\mu,\lambda}^{\mu^*}$. Now, for all $\mu \in \mathcal{M}$ and $\lambda \in \Lambda$, we define the best response operator $F : \mathcal{M} \times \Lambda \rightarrow \mathcal{M} \times \Lambda$ according to $F(\mu, \lambda) = (F_1(\mu, \lambda), F_2(\mu, \lambda))$, where

$$F_1(\mu, \lambda) = \mu_{\mu,\lambda}^*,$$

$$F_2(\mu, \lambda)(s) = \beta E_{\mu,\lambda} [V_{\mu,\lambda}^*(x^e, s_{t+1}) | s_t = s], \forall s \in \mathcal{S}^e.$$

A fixed point of the operator F 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

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

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

Step (4) in the algorithm requires solving a dynamic programming problem to optimize incumbent firms' strategies. This 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.

To formalize this idea and assess the accuracy of our approximation we introduce the following definitions. Suppose that the Markov process that describes the industry evolution if all incumbent firms use strategy μ and potential entrants use strategy λ , $\{s_t : t \geq 0\}$, admits a unique invariant distribution, which we call $q_{\mu,\lambda}$.⁸ Let $\hat{q}_{\mu,\lambda}(x, s) = q_{\mu,\lambda}(s) / |\{x : s(x) > 0\}|$, that is, a distribution induced over \mathcal{Y} , such that

⁸Under mild technical conditions such an invariant distribution exists and is unique.

an industry state s is chosen with probability $q_{\mu,\lambda}(s)$ and an individual firm's state x is chosen uniformly among all individual states for which $s(x) > 0$. Finally, let $\tilde{q}_{\mu,\lambda}(s) = q_{\mu,\lambda}(s)\mathbf{1}[s \in \mathcal{S}^e] / (\sum_{s \in \mathcal{S}^e} q_{\mu,\lambda}(s))$, where $\mathbf{1}[\cdot]$ is the indicator function. Hence, $\tilde{q}_{\mu,\lambda}$ is the conditional invariant distribution of $\{s_t : t \geq 0\}$, conditioned on $\sum_x s_t(x) < N$ (that is, on industry states that have potential entrants).

Definition 3.1. For $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, we call $(\tilde{\mu}, \tilde{\lambda}) \in \mathcal{M} \times \Lambda$ an ϵ -weighted best response to (μ, λ) if⁹

$$(3.1) \quad \|V_{\mu,\lambda}^* - V_{\mu,\lambda}^{\tilde{\mu}}\|_{1,\hat{q}_{\mu,\lambda}} \leq \epsilon,$$

$$(3.2) \quad \left\| \tilde{\lambda} - \beta E_{\mu,\lambda} \left[V_{\mu,\lambda}^{\tilde{\mu}}(x^e, s_{t+1}) | s_t = \cdot \right] \right\|_{1,\tilde{q}_{\mu,\lambda}} \leq \epsilon.$$

Definition 3.2. $(\tilde{\mu}, \tilde{\lambda}) \in \mathcal{M} \times \Lambda$ is an ϵ -weighted MPE if $(\tilde{\mu}, \tilde{\lambda})$ is an ϵ -weighted best response to itself.

Under our definition, the maximum potential gain to an incumbent firm in deviating from an ϵ -weighted MPE strategy, $\tilde{\mu}$, is averaged across industry states, with an emphasis on states visited frequently under $(\tilde{\mu}, \tilde{\lambda})$; this average gain can be at most ϵ . 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 is at most ϵ . When resorting to our approximation it will be unlikely that one can approximate the value function and strategies accurately in the entire state space. As we describe later, our efforts will be focused on states that are relevant, in the sense that they have a substantial probability of occurrence under the invariant distribution. This motivates the definition of ϵ -weighted MPE. The notion of ϵ -weighted MPE, that focuses on 'relevant' states, is similar to other concepts that have been previously used to assess the accuracy of approximations to MPE (Weintraub, Benkard, and Van Roy 2008), as stopping criteria (Pakes and McGuire 2001), and as alternative equilibrium concepts to MPE (Fershtman and Pakes 2009).

In Section 4 we will replace the operator F in Algorithm 1 by another operator \tilde{F} , which does not aim to solve for a best response, but is computationally tractable and computes a good approximation to the best response. At the i th stage of such an algorithm, we will compute $(\mu_{i+1}, \lambda_{i+1}) := \tilde{F}(\mu_i, \lambda_i)$ for which we will have that

$$(3.3) \quad \|V_{\mu_i,\lambda_i}^* - V_{\mu_i,\lambda_i}^{\mu_{i+1}}\|_{1,\hat{q}_{\mu_i,\lambda_i}} \leq \|V_{\mu_i,\lambda_i}^* - V_{\mu_i,\lambda_i}^{\mu_{i+1}}\|_{1,\hat{q}_{\mu_i,\lambda_i}} + \|V_{\mu_i,\lambda_i}^{\mu_{i+1}} - V_{\mu_i,\lambda_i}^{\lambda_{i+1}}\|_{1,\hat{q}_{\mu_i,\lambda_i}},$$

⁹For $c \in \mathbb{R}_+^k$, the $(1, c)$ norm of a vector $x \in \mathbb{R}^k$ is defined according to $\|x\|_{1,c} = \sum_{i=1}^k |x_i|c_i$.

and

$$\begin{aligned}
(3.4) \quad & \|\lambda_i - \beta E_{\mu_i, \lambda_i} [V_{\mu_i, \lambda_i}(x^e, s_{t+1}) | s_t = \cdot] \|_{1, \hat{q}_{\mu_i, \lambda_i}} \leq \|\lambda_i - \lambda_{i+1}\|_{1, \hat{q}_{\mu_i, \lambda_i}} \\
& + \|\lambda_{i+1} - \beta E_{\mu_i, \lambda_i} [V_{\mu_i, \lambda_i}^{\mu_{i+1}}(x^e, s_{t+1}) | s_t = \cdot] \|_{1, \hat{q}_{\mu_i, \lambda_i}} \\
& + \|\beta E_{\mu_i, \lambda_i} [V_{\mu_i, \lambda_i}^{\mu_{i+1}}(x^e, s_{t+1}) | s_t = \cdot] - \beta E_{\mu_i, \lambda_i} [V_{\mu_i, \lambda_i}(x^e, s_{t+1}) | s_t = \cdot] \|_{1, \hat{q}_{\mu_i, \lambda_i}}.
\end{aligned}$$

The upper bounds on the terms in (3.3) and (3.4) will form the basis of a stopping criterion for our algorithm that computes ϵ -weighted MPE. To see this, we note that if \tilde{F} were guaranteed to compute an $\epsilon/2$ -weighted best response, we would have that $\|V_{\mu_i, \lambda_i}^* - V_{\mu_i, \lambda_i}^{\mu_{i+1}}\|_{1, \hat{q}_{\mu_i, \lambda_i}} < \epsilon/2$. Moreover the last term on the right hand side of (3.3) can be estimated without knowledge of μ^* so that if we select as a stopping criterion the requirement that $\|V_{\mu_i, \lambda_i}^{\mu_{i+1}} - V_{\mu_i, \lambda_i}\|_{1, \hat{q}_{\mu_i, \lambda_i}}$ be sufficiently small (say, $< \epsilon/2$), then we would have that upon convergence, equation (3.1) for an ϵ -weighted MPE is satisfied. In addition, (3.4) does not depend on μ^* . Hence, if we update the entry rate function in our algorithm according to $\lambda_{i+1} = \beta E_{\mu_i, \lambda_i} [V_{\mu_i, \lambda_i}^{\mu_{i+1}}(x^e, s_{t+1}) | s_t = \cdot]$ and select as a stopping criterion the requirement that the first and last terms in the right hand side of (3.4) be sufficiently small (say, $< \epsilon/2$ each), then we would have that upon convergence, equation (3.2) for an ϵ -weighted MPE is satisfied. Therefore, upon convergence we obtain an ϵ -weighted MPE.

Notice that our ability to compute ϵ -weighted MPE as described above relies crucially on the quality of the approximation we can provide to the incumbents' best response at each stage of the algorithm, that is, the magnitude of $\|V_{\mu_i, \lambda_i}^* - V_{\mu_i, \lambda_i}^{\mu_{i+1}}\|_{1, \hat{q}_{\mu_i, \lambda_i}}$. This quantity is not easy to measure directly. Instead, Section 4 provides theoretical conditions under which $\|V_{\mu_i, \lambda_i}^* - V_{\mu_i, \lambda_i}^{\mu_{i+1}}\|_{1, \hat{q}_{\mu_i, \lambda_i}}$ is guaranteed to be small when the approximate best response operator, \tilde{F} , is computed using a certain linear programming algorithm. In particular, that section will show that if we are given a set of *basis functions* such that the optimal value function corresponding to a best response is 'well' approximated by a linear combination of these basis functions, then $\|V_{\mu_i, \lambda_i}^* - V_{\mu_i, \lambda_i}^{\mu_{i+1}}\|_{1, \hat{q}_{\mu_i, \lambda_i}}$ will also be small in a proportionate sense that we will make precise later. Section 4 thus lends theoretical support to the claim that the algorithm we will propose does indeed compute ϵ -weighted MPE, where ϵ depends on the 'expressiveness' of the set of given basis functions. As we improve our approximation to the best response (e.g., by adding more basis functions) we are able to find ϵ -weighted MPE with smaller values of ϵ , and as ϵ converges to zero the corresponding strategies permit vanishingly small gains to deviating in states that have positive probability of occurrence under the invariant distribution. Moreover, under one additional assumption we can prove that the strategies we converge to indeed approach a MPE.

Assumption 3.1. 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.

Assumption 3.1 together with the fact that the state space is finite, imply that the Markov chain that describes the industry state evolution $\{s_t : t \geq 0\}$ admits a unique invariant distribution. Moreover, the invariant distribution assigns strictly positive mass to all possible states.¹⁰ 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 whose proof may be found in the appendix.

Theorem 3.1. Suppose Assumption 3.1 holds. 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. Suppose that $\lim_{n \rightarrow \infty} \epsilon_n = 0$. Then, $\lim_{n \rightarrow \infty} \mathcal{D}(\Gamma, (\mu_n, \lambda_n)) = 0$.

4 Approximate Linear Programming

In Section 3 we introduced a best response algorithm to compute MPE and argued that attempting to compute a best response at every step is computationally infeasible for many applications of interest. We also developed a notion of ϵ -weighted MPE for which it sufficed to compute approximations to the best response. With this motivation in mind, this section describes how one might construct an operator \tilde{F} that computes an approximation to the best response at each step, but is computationally feasible.

In section 4.1 we specialize Algorithm 1 by performing step (4) using the linear 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 is intractable in many applications. In particular, after appropriate discretization, the best response is found by solving a linear program for which the number of variables and constraints are at least as large as the size of the state space.

Following de Farias and Van Roy (2003) and de Farias and Van Roy (2004), we alleviate the computational burden in two steps. First, in section 4.2 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

¹⁰The 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, $\bar{\kappa}$ is large enough, and $\pi(x, s) \geq 0$; and (ii) exit and entry cut-off values are restricted to belong to the sets $[0, \max_{x,s} \pi(x, s)/(1-\beta) + \kappa']$ and $[\kappa', \infty)$, respectively, where κ' 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. In our numerical experiments we assume a model that satisfies this assumption. In particular, we assume that in all states, there are exogenous depreciation and appreciation noises.

program. However, the number of constraints is still prohibitive. In section 4.3 we describe how a constraint sampling scheme alleviates this difficulty.

We will present approximation bounds that guarantee that by enriching our approximation architecture, we can produce ϵ -weighted best responses with smaller values of ϵ . By our discussion in the preceding section, this implies that upon termination the algorithm will have computed an ϵ -weighted MPE with a smaller ϵ . This in turn will yield a better approximation to a MPE strategy in the sense of Theorem 3.1. We note that such approximation bounds 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.

4.1 A Linear Programming Approach to MPE

For some $(\mu, \lambda) \in \mathcal{M} \times \Lambda$, consider the problem of finding a best response strategy $\mu_{\mu, \lambda}^*$. We will proceed in three steps:

1. We write down a ‘standard’ optimization problem for this task that is non-linear. This non-linear program is computationally difficult to solve.
2. We discretize the sell off value distribution and potential investment levels and show that the resulting problem is a good approximation.
3. We show that the standard optimization problem in step 1 can now be written as a finite linear program (LP). This LP is substantially easier to solve than the original non-linear program.

To begin, let us define for an arbitrary $\mu' \in \mathcal{M}$, the continuation value operator $C_{\mu, \lambda}^{\mu'}$ according to:

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

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

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

where the maximum is achieved point-wise. Define the operator $T_{\mu, \lambda}^{\mu'}$ according to

$$T_{\mu, \lambda}^{\mu'} V(x, s) = \pi(x, s) + \mathcal{P}[\kappa \geq \rho'(x, s)] E[\kappa | \kappa \geq \rho'(x, s)] + \mathcal{P}[\kappa < \rho'(x, s)] C_{\mu, \lambda}^{\mu'} V(x, s),$$

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

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

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

$$T_{\mu,\lambda}^{\mu^*} V_{\mu,\lambda}^* = T_{\mu,\lambda} V_{\mu,\lambda}^*.$$

A solution to Bellman's equation may be obtained via a number of algorithms. One algorithm requires us to solve the following, simple to state mathematical program:

$$(4.1) \quad \begin{aligned} \min \quad & c'V \\ \text{s.t.} \quad & (T_{\mu,\lambda}V)(x, s) \leq V(x, s), \quad \forall (x, s) \in \mathcal{Y}. \end{aligned}$$

It is a well known fact that when c is component-wise positive, the above program yields as its optimal solution the value function associated to a best response to (μ, λ) , $V_{\mu,\lambda}^*$ (Bertsekas 2001). This program has *non-linear* constraints; what follows is a discretization of the sell-off value distribution and investment levels that will allow for a useful *linear* formulation which is much simpler to solve.

4.1.1 Discretizing Sell-off Values

For given $n \in \mathbb{N}$, let us assume we are given a set $\hat{\mathcal{K}} = \{k_1, k_2, \dots, k_n\}$, satisfying the following property:

$$(4.2) \quad \left| \frac{1}{n} \sum_{i=1}^n (k_i \vee C) - E[\kappa \vee C] \right| \leq \epsilon_n, \quad \forall C \in [0, \bar{\pi}/(1 - \beta) + \bar{\kappa}],$$

in which $\epsilon_n \rightarrow 0$. There are several ways of accomplishing the above task; one concrete scheme selects k_j as the largest quantity satisfying $\Pr[\kappa < k_j] \leq \frac{n+1-j}{n}$ for $j = 1, \dots, n$. For $\hat{\mathcal{K}}$ so constructed, we will have that (4.2) is satisfied with $\epsilon_n = \bar{\kappa}/n$. We will consider solving a best-response problem assuming that sell-off values are drawn uniformly at random from $\hat{\mathcal{K}}$; we denote by $\hat{\kappa}$ such a random variable.

Given the discretization above, consider approximating the operator $T_{\mu,\lambda}$ according to:

$$(T_{\mu,\lambda}V)(x, s) \sim \pi(x, s) + \frac{1}{n} \sum_{i=1}^n (k_i \vee (C_{\mu,\lambda}V)(x, s)) \triangleq (T_{\mu,\lambda}^{\text{emp},n}V)(x, s).$$

We consider solving the following program instead of (4.1):

$$(4.3) \quad \begin{aligned} \min \quad & c'V \\ \text{s.t.} \quad & (T_{\mu,\lambda}^{\text{emp},n}V)(x, s) \leq V(x, s), \quad \forall (x, s) \in \mathcal{Y}. \end{aligned}$$

An optimal solution to the above program yields the value of an optimal best response to (μ, λ) when a firm is faced with a random sell-off value in every period, distributed uniformly in $\hat{\mathcal{K}}$. Let us call this value $\hat{V}_{\mu,\lambda}^*$ and show next that $\hat{V}_{\mu,\lambda}^*$ constitutes a good approximation to $V_{\mu,\lambda}^*$. In particular, we have the following result whose proof may be found in the appendix.

Lemma 4.1. *Let $\hat{V}_{\mu,\lambda}^*$ be an optimal solution to (4.3). We have:*

$$\|\hat{V}_{\mu,\lambda}^* - V_{\mu,\lambda}^*\|_{\infty} \leq \epsilon_n / (1 - \beta).$$

4.1.2 Discretized Investment Levels

The previous section showed that by considering an appropriate discretization of the sell-off value distribution, one may still compute a good approximation to the value of a best response to (μ, λ) . This section will show that by an appropriate discretization of investment levels, we can continue to maintain a good approximation to the best response. This will eventually allow us to consider a linear program for the computation of a near-best response to (μ, λ) . In particular, let us define for arbitrary $\epsilon > 0$, the set

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

and with a minor abuse of notation define a “discretized” Bellman operator $T_{\mu,\lambda}^{\epsilon}$ according to

$$(4.4) \quad (T_{\mu,\lambda}^{\epsilon}V)(x, s) = \max_{\mu'(x,s): \nu'(x,s) \in \mathcal{I}^{\epsilon}} (T_{\mu,\lambda}^{\mu'}V)(x, s), \quad \forall (x, s) \in \mathcal{Y}.$$

Let us denote by $V_{\mu,\lambda}^{*,\epsilon}$ the value function corresponding to a best response investment/exit strategy to (μ, λ) when investments are restricted to the set \mathcal{I}^{ϵ} . 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 $V_{\mu,\lambda}^{*,\epsilon}$. The value

function $V_{\mu}^{*,\epsilon}$ is the unique fixed point of the discretized Bellman operator $T_{\mu,\lambda}^{\epsilon}$ and may be computed by the solution of the following program:

$$(4.5) \quad \begin{aligned} \min \quad & c'V \\ \text{s.t.} \quad & (T_{\mu,\lambda}^{\epsilon}V)(x, s) \leq V(x, s) \quad \forall (x, s) \in \mathcal{Y}. \end{aligned}$$

Along with a discretization of sell-off values, the above program may be re-written as a linear program. Before doing this, we show that $V_{\mu,\lambda}^{*,\epsilon}$ is for a fine enough discretization, a good approximation to $V_{\mu,\lambda}^*$. The proof can be found in the appendix.

Lemma 4.2. *Let $\tilde{\epsilon} < 1$ satisfy $1 - \tilde{\epsilon} \leq \frac{\mathcal{P}(x_1=x'|x_0=x, \iota_0=\lfloor \iota/\epsilon \rfloor \epsilon)}{\mathcal{P}(x_1=x'|x_0=x, \iota_0=\iota)}$, $\forall x, x', \iota$. Let $V_{\mu,\lambda}^{*,\epsilon}$ be an optimal solution to (4.5). Then:*

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

Note that a finer discretization leads to smaller values of ϵ and $\tilde{\epsilon}$. We next show that discretizing the sell-off value distribution along with the set of potential investment levels, allows us to compute a good approximation to $V_{\mu,\lambda}^*$ via a linear program.

4.1.3 The Linear Program

Consider the problem of computing a best-response to (μ, λ) assuming the deviating firm faces a *discrete* sell-off value distribution that is uniform over a set $\{k_1, k_2, \dots, k_n\}$ and considers investments from some *finite* set of investment levels, \mathcal{I}^{ϵ} ; we claim that such a best response can be computed via a linear program. In particular, for a given state $(x, s) \in \mathcal{Y}$, consider the constraint

$$(T_{\mu,\lambda}V)(x, s) \leq V(x, s).$$

By introducing auxiliary variables $u(x, s) \in \mathbb{R}^n$, this constraint is equivalent to the following set of constraints:¹¹

$$(4.6) \quad \begin{aligned} \pi(x, s) + \frac{1}{n} \sum_{i=1}^n u(x, s)_i &\leq V(x, s) \\ \max_{\mu'(x,s): \iota'(x,s) \in \mathcal{I}^{\epsilon}} C_{\mu,\lambda}^{\mu'} V(x, s) &\leq u(x, s)_i \quad \forall i \in \{1, \dots, n\} \\ k_i &\leq u(x, s)_i \quad \forall i \in \{1, \dots, n\}. \end{aligned}$$

These constraint, except for the second one, are linear in the set of variables u and V . However, it is

¹¹By 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 (4.6).

easy to see that the second non-linear constraint above is equivalent to a set of $|\mathcal{T}^\epsilon|$ linear constraints:

$$-d\iota'(x, s) + \beta E_{\mu, \lambda}[V(x_1, s_1)|x_0 = x, s_0 = s, \iota_0 = \iota'(x, s)] \leq u(x, s)_i, \forall \iota'(x, s) \in \mathcal{T}^\epsilon.^{12}$$

Hence, in this setting, (4.1) is, in fact equivalent to a linear program. If the sell-off value distribution we face is not discrete, or the investment levels allowed not finite, the previous sections show how we might overcome such obstacles with discretization while maintaining a good approximation. In particular, the value of this discretized program, say $\tilde{V}_{\mu, \lambda}^*$, must satisfy by Lemmas 4.1 and 4.2, $\|\tilde{V}_{\mu, \lambda}^* - V_{\mu, \lambda}^*\|_\infty \leq \frac{\tilde{\epsilon}\beta(\bar{\pi} + \bar{\iota} + \bar{\kappa})}{(1-\beta)^2} + \frac{d\epsilon}{1-\beta} + \epsilon_n$ where the quantities ϵ , $\tilde{\epsilon}$ and ϵ_n are described in the statement of those results, and can be made arbitrarily small by an appropriate discretization.

In spite of the simplification achieved via our discretization, (4.1) remains an intractable program. In particular, the equivalent linear program has as many variables as the size of the state space $|\mathcal{Y}|$, and at least that many constraints; as we have discussed $|\mathcal{Y}|$ is likely to be a very large quantity for problems of interest. We next focus on computing a good ‘approximate’ solution to this intractable program by restricting attention to approximations of the value function and using a constraint sampling procedure that will be the subject of the next two sections.

4.2 Value Function Approximation

For the remainder of this section, with a view to avoiding cumbersome notation, we will understand that the deviating firm has a discrete valued sell-off distribution of the type discussed in the previous section, and is allowed a discrete set of investments. Assume we are given a set of “basis” functions $\Phi_i : \mathcal{Y} \rightarrow \mathbb{R}$, for $i = 1, 2, \dots, K$. Let us denote by $\Phi \in \mathbb{R}^{|\mathcal{Y}| \times K}$ the matrix $[\Phi_1, \Phi_2, \dots, \Phi_K]$. 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:

$$(4.7) \quad \begin{aligned} \min \quad & c' \Phi r \\ \text{s.t.} \quad & (T_{\mu, \lambda} \Phi r)(x, s) \leq (\Phi r)(x, s) \quad , \forall (x, s) \in \mathcal{Y}. \end{aligned}$$

As discussed in the previous section, the above program can be rewritten as a linear program via (4.6). The above program attempts to find a good approximation to $V_{\mu, \lambda}^*$ within the linear span of the basis functions $\Phi_1, \Phi_2, \dots, \Phi_K$. The idea is that if the basis functions are selected so that they can closely approximate the value function $V_{\mu, \lambda}^*$, then the program (4.7) should provide an effective approximation. The following

¹²Note that for a fixed action $\iota'(x, s)$ the expectation operator is linear in the set of variables V .

Theorem formalizes this notion:

Theorem 4.1. *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 (4.7). Then,*

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

The result follows via an argument essentially identical to Theorem 2 of de Farias and Van Roy (2003); the proof is omitted. 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). The number of constraints that we must contend with continues to remain large, and in linearizing each of these constraints via (4.6) we will introduce as many new auxiliary variables; we will eventually resort to a constraint sampling scheme to compute a solution to this program. This issue notwithstanding, (4.7) represents a substantial simplification to the program (4.1) we started with.

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

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

Provided $\Phi r_{\mu,\lambda}$ is a good approximation to $V_{\mu,\lambda}^*$, the expected discounted payoffs 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 $q_{\mu,\lambda}^{\mu'}$ the sub-probability distribution

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

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

Theorem 4.2. *Given strategies (μ, λ) , and defining $\tilde{\mu}$ and $q_{\mu,\lambda}^{\tilde{\mu}}$ as above for an arbitrary distribution over*

initial states in \mathcal{Y} , ν , we have:

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

Together with Theorem 4.1, this result lets us conclude that

$$(4.8) \quad \|V_{\mu,\lambda}^{\tilde{\mu}} - V_{\mu,\lambda}^*\|_{1,\nu} \leq \max_{(x,s) \in \mathcal{Y}} \frac{q_{\mu,\lambda}^{\tilde{\mu}}(x,s)}{c(x,s)} \frac{2}{(1-\beta)^2} \inf_r \|\Phi r - V_{\mu,\lambda}^*\|_{\infty}.$$

It is worth pausing to discuss what we have established thus far. Assume we could solve the program (4.7) and thus compute an approximate best response $\tilde{\mu}$ to (μ, λ) at every stage of Algorithm 1. If we assume $\nu = \hat{q}_{\mu,\lambda}$, $\tilde{\mu}$ would then be an ϵ -weighted best response with ϵ specified by the right hand side of (4.8). In particular, assume that upon convergence our iterative best response scheme converged to some investment/exit strategy and entry strategy $(\bar{\mu}, \bar{\lambda})$. Let $\tilde{\mu}$ be an approximate best response to $(\bar{\mu}, \bar{\lambda})$. Then, by (3.3), $(\bar{\mu}, \bar{\lambda})$ would satisfy equation (3.1) of an ϵ -weighted MPE with

$$(4.9) \quad \epsilon = \max_{(x,s) \in \mathcal{Y}} \frac{q_{\bar{\mu},\bar{\lambda}}^{\tilde{\mu}}(x,s)}{c(x,s)} \frac{2}{(1-\beta)^2} \inf_r \|\Phi r - V_{\bar{\mu},\bar{\lambda}}^*\|_{\infty} + \epsilon',$$

where ϵ' can be made arbitrarily small by selecting an appropriate stopping criterion and we assume that $\nu = \hat{q}_{\bar{\mu},\bar{\lambda}}$. This expression highlights the drivers of our ability to compute good approximations to MPE. In particular, these are:

1. Our ability to approximate the optimal value function when competitors use the candidate equilibrium strategy within the span of the chosen basis functions. In particular, as we improve the approximation architecture (for example, by adding basis functions), we see from the above expression that we are capable of producing ϵ -weighted MPE for smaller ϵ . This in turn will yield better approximations to MPE in the sense of Theorem 3.1.
2. The state relevance weight vector c plays the role of trading off approximation error across states which follows from the fact that (4.7) is equivalent to the program (see de Farias and Van Roy (2003)):

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

As suggested by (4.9), the vector c should ideally assign weights to industry states according to the invariant distribution of the Markov process that describes the industry evolution when firms use the

candidate equilibrium strategy.

4.3 Reducing the Number of Constraints

The previous section reduced the problem of finding an approximate best response to an incumbent investment/exit strategy and entry cutoff (μ, λ) to the solution of a linear program with a large number of variables and constraints (4.7). While that program reduced the number of variables via value function approximation, the number of auxiliary variables needed to linearize that program is as large as the number of constraints. This section focuses on developing a practical scheme to approximately solve such a program. In particular, 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 (4.7) 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 (4.7):

$$(4.10) \quad \begin{aligned} \min \quad & c' \Phi r \\ \text{s.t.} \quad & (T_{\mu, \lambda} \Phi r)(x, s) \leq (\Phi r)(x, s) \quad \forall (x, s) \in \mathcal{R}. \end{aligned}$$

Intuitively, a sufficiently large number of samples L should suffice to guarantee that a solution to (4.10) satisfies all except a small set of constraints in (4.7) with high probability. In fact, it has been shown in de Farias and Van Roy (2004) that for a specialized choice of the sampling distribution, L can be chosen independently of the total number of constraints in order to achieve a desired level of performance. In particular, assume we had access to the strategy $\mu_{\mu, \lambda}^*$ and consider sampling states according to a distribution ψ^* defined according to $\psi^*(x, s) = q_{\mu, \lambda}^{\mu^*}(x, s) / \sum_{(x', s') \in \mathcal{Y}} q_{\mu, \lambda}^{\mu^*}(x', s')$ where we take ν to be equal to the state relevant weights vector, c , in the definition of $q_{\mu, \lambda}^{\mu^*}$. We do not have access to ψ^* ; let $\bar{\psi}$ be a sampling distribution satisfying $\max_{x, s} \frac{\psi^*(x, s)}{\psi(x, s)} \leq M$. Assuming the L states in \mathcal{R} are sampled according to $\bar{\psi}$, we then have the following result, specialized from de Farias and Van Roy (2004) and whose proof is omitted:

Theorem 4.3. *Let $\delta, \epsilon' \in (0, 1)$. Let \mathcal{R} consist of L states in \mathcal{Y} sampled according to $\bar{\psi}$. Let $\tilde{r}_{\mu, \lambda}$ be an optimal solution to (4.10). If*

$$L \geq \frac{16 \|V_{\mu, \lambda}^* - \Phi \tilde{r}_{\mu, \lambda}\|_{\infty} M}{(1 - \beta) \epsilon' c^{\top} V_{\mu, \lambda}^*} \left(K \ln \frac{48 \|V_{\mu, \lambda}^* - \Phi \tilde{r}_{\mu, \lambda}\|_{\infty} M}{(1 - \beta) \epsilon' c^{\top} V_{\mu, \lambda}^*} + \ln \frac{2}{\delta} \right),$$

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 (4.7).

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 $\|V_{\mu,\lambda}^* - \Phi \tilde{r}_{\mu,\lambda}\|_{1,c} \approx \|V_{\mu,\lambda}^* - \Phi r_{\mu,\lambda}\|_{1,c}$.¹³ 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 $\bar{\psi}$: in every iteration we simply sample industry states according to the invariant distribution of the Markov process that describes the industry evolution when firms use the approximate best response strategy computed at the prior iteration.

Theorems 4.1, 4.2 and 4.3 together establish the quality of our approximation to a best response computed via a *tractable* linear program (4.10). In particular, we showed that by sampling a sufficiently large, but tractable, number of constraints via an appropriate sampling distribution, one could compute an approximate best response whose quality is similar to that of an approximate best response computed via the intractable linear program (4.7); the quality of that approximate best response was established in the preceding section.

This section established a tractable computational scheme to compute an approximation to the best response operator $F(\cdot)$ in step 4 of Algorithm 1: in particular, we suggest approximating $F(\mu, \lambda)$ by a strategy $\tilde{\mu}$ satisfying $T_{\mu,\lambda}^{\tilde{\mu}} \Phi \tilde{r}_{\mu,\lambda} = T_{\mu,\lambda} \Phi \tilde{r}_{\mu,\lambda}$ where $\tilde{r}_{\mu,\lambda}$ is an optimal solution to the tractable LP (4.10). A number of details, such as the choice of basis functions Φ , and the precise way the state relevant weights c and sampling distribution $\bar{\psi}$ are determined, remain unresolved. These will be addressed in subsequent sections where we describe our algorithm (in the format of a ‘guide to implementation’) to compute an approximation to MPE precisely, and discuss our computational experiments. The implementation issues that we discuss are key to the efficiency of our algorithm and the accuracy of our approximation.

5 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 5.1. The following sections

¹³The number grows linearly in the number of basis functions and is independent from the total number of constraints.

present important sub-routines. Theoretical support for the procedure described here has been presented in Section 4.

5.1 Algorithm

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

- $\{\Phi_i : i = 1, \dots, K\}$, a collection of K basis functions. This collection is such that $\Phi_i : \mathcal{Y} \rightarrow \mathbb{R}$ for all $i = 1, 2, \dots, K$. We denote by $\Phi \in \mathbb{R}^{|\mathcal{Y}| \times K}$ the matrix $[\Phi_1, \Phi_2, \dots, \Phi_K]$.
- A discrete valued random variable \hat{k} taking values uniformly at random in a set $\{k_i : i \in \mathcal{N}\}$ where \mathcal{N} 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.
- A discrete set of permissible investment levels $\mathcal{I} = \{0, \epsilon, 2\epsilon, \dots, \lfloor \bar{l}/\epsilon \rfloor \epsilon\}$. Again, \mathcal{I} may be viewed as a discretization of some given set of permissible investment levels.
- (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:

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

- An arbitrary initial state in \mathcal{Y} , v .
- 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 steps (4) and (5) in Algorithm 2 and (5.1)); *ii*) T , a positive integer that determines the size of the transient period when simulating the industry evolution; and *iii*) $\epsilon > 0$ used to calibrate the exit criteria (see step (13) in Algorithm 2).

We next describe our Algorithm, noting that the description will call on two procedures we are yet to describe, namely, a linear programming sub-routine $ALP(\cdot)$, and an oracle $M(\cdot)$ that succinctly represents investment, entry and exit strategies using the results of previous iteration.

Upon convergence, the approximate MPE strategy can be recovered as $(\mu_{i^*}, \lambda_{i^*}) = M(r^{i^*}, \dots, r^1, \mu^c, \lambda^c)$, where i^* is the number of iterations until convergence, and r^j are the weights obtained in $ALP(\cdot)$ at each

Algorithm 2 Algorithm for Approximating MPE

1: $\mu_0 = \mu^c, \lambda_0 = \lambda^c$

{Set initial investment, entry and exit strategies}

2: $i := 0$

{ i indexes best response iterations}

3: **repeat**

4: Simulate industry evolution over $T + \tilde{L}$ periods, assuming all firms use strategy (μ_i, λ_i) and the initial industry state is v .

- Compute empirical distribution q .

$$q(x, s) = \sum_{t=T}^{T+\tilde{L}} \mathbf{1}[s_t(x) > 0, s_t = s] / \sum_{t=T}^{T+\tilde{L}} \sum_{x \in \mathcal{X}} \mathbf{1}[s_t(x) > 0], \forall (x, s) \in \mathcal{Y}$$

- $\mathcal{R} \leftarrow \{(x, s) \in \mathcal{Y} : q(x, s) > 0\}$. Let $L = |\mathcal{R}|$.

{The distribution q is the empirical counterpart of $\hat{q}_{\mu_i, \lambda_i}$. q and \mathcal{R} are used to build the ALP; see Section 4 for the theoretical justification.}

5: Set $r^{i+1} \leftarrow ALP(\mathcal{R}, \mu_i, \lambda_i, q)$

{The $ALP(\cdot)$ procedure produces an approximate best response to (μ_i, λ_i) ; this is succinctly described by the parameter vector r^{i+1} . See the following Section for the description of $ALP(\cdot)$ }

6: $(\mu_{i+1}, \lambda_{i+1}) := M(r^{i+1}, \dots, r^1, \mu_0, \lambda_0)$.

{The oracle $M(\cdot)$ described in Section 5.3 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.}

7: **for each** $(x, s) \in \mathcal{R}$ **do**

8: Estimate $V_{\mu_i \lambda_i}^{\mu_{i+1}}(x, s)$.

{Estimation is via monte-carlo simulation of industry evolution starting from (x, s) with the incumbent firm using strategy μ_{i+1} and its competitors using (μ_i, λ_i) .}

9: Estimate $V_{\mu_i \lambda_i}(x, s)$.

{Estimation is via monte-carlo simulation of industry evolution starting from (x, s) with all firms using strategy (μ_i, λ_i) .}

10: **end for**

11: $\Delta = \sum_{(x,s) \in \mathcal{R}} q(x, s) \left| V_{\mu_i \lambda_i}^{\mu_{i+1}}(x, s) - V_{\mu_i \lambda_i}(x, s) \right|$

{ Δ is the empirical counterpart to the directly measurable component of the quality of a candidate equilibrium; see the second term of (3.3).}

12: $i := i + 1$

13: **until** $\Delta < \epsilon$

iteration. In our computational experiments we allow for a ‘smooth’ update of the r values. Specifically we performed the update $r_i := r_{i-1} + (r_i - r_{i-1})/i^\gamma$ as a last step in every iteration.¹⁴ Intuition for the selected stopping criteria is given in Section 3.¹⁵ The above algorithmic description has left two ‘sub-routines’ unspecified, these are the linear program $ALP(\cdot)$, and the oracle $M(\cdot)$. The next two sections are devoted to describing these sub-routines in detail.

5.2 The Linear Programming Sub-routine $ALP(\cdot)$

The $ALP(\mathcal{R}, \mu, \lambda, q)$ sub-routine employed in step 5 of Algorithm 2 simply outputs the solution of the following linear program:¹⁶

(5.1)

$ALP(\mathcal{R}, \mu, \lambda, q) :$

$$\begin{aligned}
& \underset{r, t, u}{\text{minimize}} && \sum_{(x, s) \in \mathcal{R}} q(x, s) \sum_{0 \leq j \leq K} \Phi_j(x, s) r_j \\
& \text{subject to} && \pi(x, s) + \frac{1}{n} \sum_{i=1}^n u(x, s)_i \leq \sum_{0 \leq j \leq K} \Phi_j(x, s) r_j && \forall (x, s) \in \mathcal{R} \\
& && -dt + \beta E_{\mu, \lambda} \left[\sum_{0 \leq j \leq K} \Phi_j(x_1, s_1) r_j \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right] \leq t(x, s) && \forall (x, s) \in \mathcal{R}, \iota \in \mathcal{I} \\
& && t(x, s) \leq u(x, s)_i && \forall (x, s) \in \mathcal{R}, i \in \mathcal{N} \\
& && k_i \leq u(x, s)_i && \forall (x, s) \in \mathcal{R}, i \in \mathcal{N}
\end{aligned}$$

The above LP is essentially identical to the program (4.10), that was developed and analyzed in Section 4, via the linearization (4.6).¹⁷ Briefly, we recall that by the theoretical justification provided in Section 4, 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 .

The second set of constraints in (5.1) involve the expectations $E_{\mu, \lambda} [(\Phi r)(x_1, s_1) | x_0 = x, s_0 = s, \iota_0 = \iota]$. Under the model and set of basis functions we introduce in section 6.1, these expectations can themselves be expressed tractably. That is, these expectations can be written as linear functions in r whose coefficients may be computed with roughly $|\mathcal{X}|N^4$ operations.¹⁸ This may not be possible in other models or with other

¹⁴The parameter γ was set after some experimentation equal to $2/3$ to speed up convergence.

¹⁵In our computational experiments we relaxed the stopping criteria discussed there and only checked the empirical counterpart of the second term in (3.3).

¹⁶To preclude that the program was unbounded we also included the constraint that r lies in a large bounded set.

¹⁷This program has one additional set of auxiliary variables t that is used to reduce the total number of constraints, albeit at a modest increase in the number of variables.

¹⁸In that model, firms can only transition to adjacent individual states. Considering this and the nature of the basis functions,

types of basis functions; in that case one may simply replace the expectation with its empirical counterpart.

Problem (5.1) has roughly $(K + L(n + 1))$ decision variables and roughly $(L(|\mathcal{I}| + n + 1))$ constraints. Thus both the number of constraints and variables in (5.1) 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. However, an alternative procedure proved to provide a further speedup. We describe this procedure next.

5.2.1 A Fast Heuristic LP Solver

We present here a fast iterative heuristic for the solution of the LP (5.1) that constitutes the subroutine $ALP(\mathcal{R}, \mu, \lambda, q)$. As opposed to solving a single LP with roughly $(K + L(n + 1))$ decision variables and roughly $(L(|\mathcal{I}| + n + 1))$ constraints, the procedure solves a sequence of substantially smaller LPs, each with K variables and $L|\mathcal{I}|$ constraints. The idea underlying the heuristic is quite simple: whereas LP (5.1) 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 (5.1). 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 were derived in the first place. The fixed points of the above approach constitute an optimal solution to (5.1). In particular, suppose $\epsilon = 0$ in the specification of Algorithm 3 and let r' be the output of the Algorithm assuming it terminates. Define

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

and

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

We then have the following result that we prove in the appendix:

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

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 $j - 1$, and $j + 1$, while at the same time considering firms leaving/entering the industry.

Algorithm 3 Heuristic to solve Linear Program $ALP(\mathcal{R}, \mu, \lambda, q)$

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 $e_j(x, s) = \max_{\iota \in \mathcal{I}} -d\iota + \beta E_{\mu, \lambda} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r'_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right], \forall (x, s) \in \mathcal{R}$.

{Set cutoff values for firm exit based on the current approximation to the optimal value function.}

4: **repeat**

5: Set r' as a solution to

(5.2)

$$\underset{r}{\text{minimize}} \quad \sum_{(x, s) \in \mathcal{R}} q(x, s) \sum_{0 \leq k \leq K} \Phi_k(x, s) r_k$$

$$\text{subject to} \quad \pi(x, s) + \mathcal{P}(\hat{\kappa} < e_j(x, s)) \left(-d\iota + \beta E_{\mu, \lambda} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right] \right) \\ + E[\hat{\kappa} | \hat{\kappa} \geq e_j(x, s)] \mathcal{P}(\hat{\kappa} \geq e_j(x, s)) \leq \sum_{0 \leq k \leq K} \Phi_k(x, s) r_k,$$

$$\forall (x, s) \in \mathcal{R}, \iota \in \mathcal{I}$$

{Compute an approximate best response investment strategy assuming the *fixed* exit rule determined by the cutoff value e_j .}

6: Set $e_{j+1}(x, s) = \max_{\iota \in \mathcal{I}} -d\iota + \beta E_{\mu, \lambda} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r'_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right], \forall (x, s) \in \mathcal{R}$.

{Update the exit rule based on the computed approximate best response investment strategy.}

7: $\Delta := \sum_{(x, s) \in \mathcal{R}} q(x, s) |e_{j+1}(x, s) - e_j(x, s)|$

{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 (5.1).}

8: $j := j + 1$

9: **until** $\Delta < \epsilon$

10: **return** r'

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 (5.1) 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 (5.1) directly) provided a substantial speedup.

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.

5.3 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 (5.1) and the initial compact-representation strategy (μ^c, λ^c) (it is understood that only $\mu_i(x, s)$ is computed when $s \notin \mathcal{S}^e$).

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:

- 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 $|\mathfrak{N}(x, s)| \leq \min\{N, |\mathcal{X}|\}$.

Note that in the last iteration of Algorithm 4 in steps (4) to (5) we require, in addition to knowing r^j ,

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:

$$\iota_j(y, s) := \operatorname{argmax}_{\iota \in \mathcal{I}} -d\iota + \beta E_{(\mu_{j-1}, \lambda_{j-1})} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r_k^j \middle| x_0 = y, s_0 = s, \iota_0 = \iota \right].$$

$$\rho_j(y, s) := -d\iota_j(y, s) + \beta E_{(\mu_{j-1}, \lambda_{j-1})} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r_k^j \middle| x_0 = y, s_0 = s, \iota_0 = \iota_j(y, s) \right].$$

$$\lambda_j(s) := \beta E_{(\mu_{j-1}, \lambda_{j-1})} \left[\sum_{0 \leq k \leq K} \Phi_k(x^e, s_1) r_k^j \middle| s_0 = s \right].$$

- 5: $j := j + 1$
 - 6: **end for**
 - 7: **until** $j = i$
 - 8: **return** $\iota_i(x, s), \rho_i(x, s), \lambda_i(s)$
-

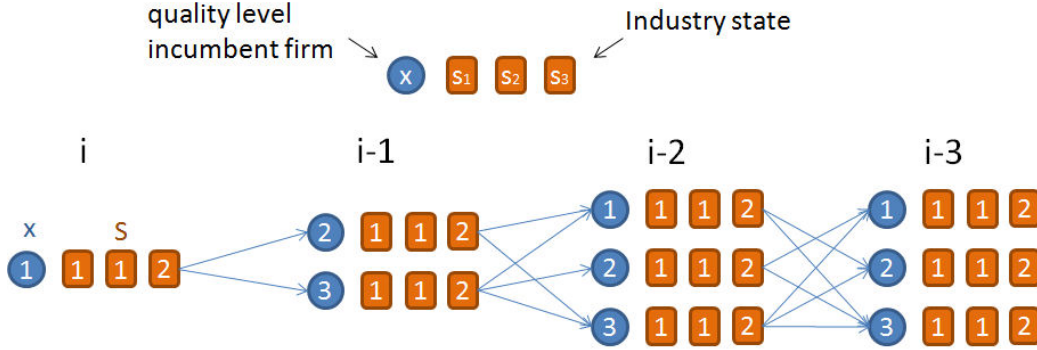


Figure 1: Oracle computation increases linearly with the number of iterates in Algorithm 2. Suppose we want to compute $\mu_i(x, s)$ and that $x = 1$ and $s = (1, 1, 2)$ (here $|\mathcal{X}| = 3$), that is, the industry state consists of one firm at quality levels 1 and 2, and 2 firms at quality level 3; the incumbent firm under consideration is in quality level 1. Computation of $\mu_i(x, s)$ requires μ_{i-1} for the competitors of the firm in quality level 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 $\mathfrak{N}(x, s)$.

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 1, prevents the computation from blowing up; since $|\mathfrak{N}(x, s)| \leq \min\{N, |\mathcal{X}|\}$, 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.

6 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 begin by specifying the EP-style model to be analyzed. The model is similar to Pakes and McGuire (1994) and Weintraub, Benkard, and Van Roy (2009). However, it differs in that we do not consider an aggregate shock that is common to all firms. Extending our approach to a model with aggregate shocks is straightforward. Then, in Section 6.2 we propose an approximation architecture for the class of models we study. Finally, in Section 6.3 we describe the benchmarks against which we compare the outcomes of our approximate dynamic programming algorithm, and we report the numerical results.

6.1 The Computational Model

SINGLE-PERIOD PROFIT FUNCTION. 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:

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

where Y is the consumer's income, p_{it} is the price of the good produced by firm i at time t , and Z is a scaling factor. ϵ_{ijt} are i.i.d. Gumbel random variables that represent unobserved characteristics for each consumer-good pair. There is also an outside good that provides consumers an average utility of zero. We

assume consumers buy at most one product each period and that they choose the product that maximizes utility. Under these assumptions our demand system is a classic logit model.

Let $R(x_{it}, p_{it}) = \exp(\theta_1 \ln(\frac{x_{it}}{Z} + 1) + \theta_2 \ln(Y - p_{it}))$. Then, the expected market share of each firm is given by:

$$\sigma(x_{it}, s_t, p_t) = \frac{R(x_{it}, p_{it})}{1 + \sum_{j \in S_t} R(x_{jt}, p_{jt})}, \forall i \in S_t.$$

We assume that firms set prices in the spot market. If there is a constant marginal cost c , the Nash equilibrium of the pricing game satisfies the first-order conditions,

$$(6.1) \quad Y - p_{it} + \theta_2(p_{it} - c)(\sigma(x_{it}, s_t, p_t) - 1) = 0, \forall i \in S_t.$$

There is a unique Nash equilibrium in pure strategies, denoted p_t^* (Caplin and Nalebuff 1991). Expected profits are given by:

$$\pi_m(x_{it}, s_t) = m\sigma(x_{it}, s_t, p_t^*)(p_{it}^* - c), \forall i \in S_t.$$

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 product depreciates one quality level 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.¹⁹ 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:

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

ENTRY AND EXIT. 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 $i \in S_t'$ will observe a random (independent) positive entry cost ϕ_{it} exponentially distributed, with mean $\tilde{\phi}$. Also, each period, each incumbent firm $i \in S_t$ observes a positive random (independent) sell-off value κ_{it} exponentially distributed, with mean $\tilde{\kappa}$.²⁰

¹⁹The appreciation shock is introduced so that Assumption 3.1 is satisfied. In our numerical experiments, the probability of going up due to this shock is chosen to be small (0.1).

²⁰Assumption 2.1 imposes bounded sell-off values and entry costs. Assuming exponentially distributed random variables simplifies the analysis. Moreover, because these random variables are 'light-tailed', in practice they can be well approximated by bounded

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 particular set of representative parameter values. Following Pakes and McGuire (1994) we fix $b = 3$, $\delta = 0.7$. Additionally, we fix income at $Y = 1$, $\theta_2 = 0.5$, and $\gamma = 0.1$. The discount factor is $\beta = 0.925$. We will keep the parameters above fixed for all experiments, unless otherwise stated. We seek to test our algorithm in situations where a MPE can be computed exactly or OE provides a reasonable approximation. Other parameters will be chosen later to accommodate the setting to one of these situations.²¹

6.2 A Separable Approximation Architecture

A key issue in our approach is the selection of an approximation architecture, that is, a set of basis functions. For the class of EP models we study, we propose using a *separable* approximation to the value of using strategy $\tilde{\mu}$ in response to strategy (μ, λ) . In particular, we use an approximation of the form:

$$V_{\mu, \lambda}^{\tilde{\mu}}(x, s) \sim f_0 + \sum_{j \in \mathcal{X}} f_x^j(s(j)),$$

where $f_x^j : \{0, \dots, N\} \rightarrow \mathbb{R}$ is an arbitrary univariate function specified for each pair $(x, j) \in \mathcal{X} \times \mathcal{X}$, and f_0 is a constant term. The approximation is separable over states $j \in \mathcal{X}$; it approximates the value function at a state (x, s) by a sum of functions of $s(j)$ for each $j \in \mathcal{X}$. Each of these basis functions only depends on the number of firms at a particular state $j \in \mathcal{X}$. More sophisticated architectures, that, for example, explicitly capture interaction effects between numbers of firms at different states, could be considered. However, this simple architecture appears to produce effective approximations in the class of EP models we study as is borne out in the computational experiments we present below.

We encode this approximation architecture using the following set of basis functions. For all $i, j \in \mathcal{X}$ and $k \in \{0, 1, \dots, N\}$, define the indicator function:

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

The basis matrix Φ is thus a matrix in $\mathbb{R}^{|\mathcal{Y}| \times |\mathcal{X}| \cdot |\mathcal{X}| \cdot (N+1)}$ whose columns are functions of the type $\Phi_{i,j,k}(\cdot, \cdot)$ random variables.

²¹To mitigate the effect of the exogenous bound \bar{x} on the dynamics, the parameters in each of the instances presented below were chosen so that the long-run expected quality level of a firm was not close to \bar{x} . In this way, the expected percentage of firms at quality level \bar{x} was always below 4.00%.

Given a vector of weights $r \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{X}| \cdot (N+1)}$, we have

$$f_x^j(s(j)) = \sum_k \Phi_{x,j,k}(x, s) r_{x,j,k} = r_{x,j,s(j)},$$

and the corresponding approximation for the value function at state (x, s) is given by

$$\sum_{j \in \mathcal{X}} f_x^j(s(j)) = (\Phi r)(x, s) = \sum_{j,k} \Phi_{x,j,k}(x, s) r_{x,j,k} = \sum_{j \in \mathcal{X}} r_{x,j,s(j)}.$$

Observe that since $|\mathcal{X}| \cdot |\mathcal{X}| \cdot (N + 1)$ will typically be substantially smaller than $|\mathcal{Y}|$, 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 millions of billions states only thousands of basis functions are required.

Note that our selection of basis functions produces the most general possible separable approximation. For each x, j , the function $f_x^j(s)$ can take different values for each different $s \in \{0, \dots, N\}$.²² Our selection of basis functions generalizes approximation architectures for which the value function is approximated as a linear combination of moments of the industry state. Moment-based approximations have been previously used in large scale stochastic control problems that arise in macroeconomics (Krusell and Smith 1998) – using a very different approach and algorithm than ours, though. In our computational experiments we observe that moving beyond simple linear combinations of, say, the first two moments of the industry state is valuable: often the simpler architecture fails to produce good approximations to MPE for our computational examples while our proposed architecture does.

6.3 Comparing Economic Indicators of Interest

We show that our approximate linear programming-based (ALP-based) algorithm with the proposed architecture provide 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

²²In our numerical experiments, we also use a (coarser) piece-wise linear separable approximation architecture. Specifically, for instances with $N \geq 20$ we introduce this architecture by modifying the linear program in Algorithm 3 as follows: For a set $\mathcal{H} \subseteq \{0, \dots, N\}$ 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.

firms and parameter regimes where OE is known to provide a good approximation.²³

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 (C_i), 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 C_2 , while if $N = 40$ one may also be interested in C_6).

6.3.1 Comparison with MPE

Exact calculation 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 quality levels 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 1.²⁴

We consider several parameter regimes. First, we consider two regimes in a model with entry and exit: 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 a model with a fixed number of firms N . Table 1 depicts parameter selection for each instance. All tables can be found in the appendix.

Table 2 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 $N = 5$.²⁵

Our approximation provides accurate approximations of the economic indicators of interest in all instances. In fact, ALP-based indicators are always within 5.5% from MPE indicators, and often within 1%. The differences are slightly larger for the models with entry and exit.

The results show that our ALP-based algorithm produces a good approximation to MPE, in instances

²³Since 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 always less than 5%.

²⁴We discretize actions and sell-off values to compute MPE.

²⁵All runs were performed on a workstation with a processor Intel(R) Xeon(R) (X5365 3.00GHz) and 32GB of RAM, in a Java implementation of the algorithm. Our Java implementation called CPLEX 11.0 as a subroutine to solve the linear programs in the algorithms.

with relatively small state spaces for which MPE can be computed. Moreover, our ALP-based algorithm requires substantially less computational effort.

6.3.2 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 (2009) provide an efficient simulation-based algorithm that computes a bound on the approximation error. Error is measured in terms of the expected incremental value that an individual firm in the industry can capture by unilaterally deviating from the OE strategy to a Markov best response. They show that the error bound is a good indicator on how accurately OE approximates MPE.

Weintraub, Benkard, and Van Roy (2009) provide extensive numerical experiments and show that OE approximate MPE better in some industries than others, depending on industry concentration and the nature of competitive interactions. In some instances, the OE approximation works well in industries with tens of firms. In others, the approximation fails to work well until there are over a hundred firms in the industry. For the purpose of our comparisons, we select parameters regimes for which OE provide accurate approximations in industries with tens of firms.

Similarly to the comparison with MPE we consider several parameter regimes that yield different investment levels, and with and with out entry and exit. Table 3 depicts parameter selection for each instance in this setting. We consider an additional parameter $\bar{m} = 10$, which serves as a base market size, such that the actual market size in the industry is $m = N\bar{m}$ (so we scale the market size proportionally to the total number of firms the industry can accommodate). For these instances, computing a MPE using Algorithm 1 is infeasible for $N > 4$. For the studied configurations, the bounds derived on Weintraub, Benkard, and Van Roy (2009) suggest that OE indicators provide a good approximation to MPE. Our ALP-based algorithm has a running time that is on the order of hours. We sampled 500,000 constraints. In our numerical experiments we solved the linear programs naively, in the sense that we did not use the structure of previous linear programs to construct the new ones that arose in the progression of the algorithm. We believe that using the structure of previous linear programs can significantly reduce the running times. This is a matter of current research.

We observe that ALP-based indicators are always within 8.5% from OE indicators, and often within 5%.

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 produces 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.

We emphasize that there is significant parameter regime for which OE is not known to be a good approximation to MPE and exact computation of MPE is not feasible. 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 this regime it is difficult to numerically test the validity of our approach since no benchmarks are available. Nonetheless, we believe that the numerical experiments described above suggest that our approximation architecture should also be capable of capturing the true value function in this regime. Moreover, with the theoretical performance guarantees we presented for our approach, this in turn suggests that upon convergence our scheme will produce effective approximations to MPE here as well. We believe our method offers the first viable approach to approximating MPE in this regime, significantly expanding the range of problems that can be analyzed.

7 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 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.

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 showed that a simple separable approximation architecture is effective for the class of EP-style models we study. There are natural extensions to this set of basis functions that may be used if a richer architecture is called for. For instance, one could consider sums of functions of coordinate pairs for all such pairs. 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.
- 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.
- 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.
- Doraszelski, U. (2003). An r&d race with knowledge accumulation. *RAND Journal of Economics* 34(1), 19 – 41.
- Doraszelski, U. and K. Judd (2006). Avoiding the curse of dimensionality in dynamic stochastic games. Working Paper, Hoover Institution.
- 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 (2007). Computable markov-perfect industry dynamics: Existence, purification, and multiplicity. Working Paper, Harvard University.
- Ericson, R. and A. Pakes (1995). Markov-perfect industry dynamics: A framework for empirical work. *Review of Economic Studies* 62(1), 53 – 82.

- Fershtman, C. and A. Pakes (2009). Finite state dynamic games with asymmetric information: A framework for applied work. Working Paper, Harvard University.
- Fudenberg, D. and J. Tirole (1991). *Game Theory*. MIT Press.
- 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.
- Perakis, G., S. Kachani, and C. Simon (2008). Closed loop dynamic pricing under competition. Working Paper, MIT.
- Pesendorfer, M. and P. Schmidt-Dengler (2003). Identification and estimation of dynamic games. Mimeo, London School of Economics.
- 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.
- 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 (2009). Computational methods for oblivious equilibrium. forthcoming, *Operations Research*.

A Proofs

Proof of Theorem 3.1. 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:

$$(A.1) \quad \|V_{\mu'_n, \lambda'_n}^{\mathcal{BR}(\mu'_n, \lambda'_n)} - V_{\mu'_n, \lambda'_n}\|_{1, \hat{q}_{\mu'_n, \lambda'_n}} \rightarrow 0,$$

$$(A.2) \quad \|\lambda'_n - \beta E_{\mu'_n, \lambda'_n} [V_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = \cdot]\|_{1, \tilde{q}_{\mu'_n, \lambda'_n}} \rightarrow 0,$$

$$(A.3) \quad (\mu'_n, \lambda'_n) \rightarrow (\mu^*, \lambda^*).$$

$$(A.4) \quad \inf_n d(\Gamma, (\mu'_n, \lambda'_n)) > \epsilon,$$

where $\mathcal{BR}(\mu, \lambda)$ denotes the best response strategy when competitors play strategy μ and enter according to λ . Now, (A.3) and Assumptions 2.1.2, 2.1.5, and 2.1.8 imply that $\hat{q}_{\mu'_n, \lambda'_n} \rightarrow \hat{q}_{\mu^*, \lambda^*}$ and $\tilde{q}_{\mu'_n, \lambda'_n} \rightarrow \tilde{q}_{\mu^*, \lambda^*}$. In addition,

$$0 \leq \|V_{\mu'_n, \lambda'_n}^{\mathcal{BR}(\mu'_n, \lambda'_n)} - V_{\mu'_n, \lambda'_n}\|_{1, \hat{q}_{\mu^*, \lambda^*}} \leq \left(\frac{\bar{\pi} + \bar{\iota}}{1 - \beta} + \bar{\kappa} \right) \|\hat{q}_{\mu^*, \lambda^*} - \hat{q}_{\mu'_n, \lambda'_n}\|_1 + \|V_{\mu'_n, \lambda'_n}^{\mathcal{BR}(\mu'_n, \lambda'_n)} - V_{\mu'_n, \lambda'_n}\|_{1, \hat{q}_{\mu'_n, \lambda'_n}},$$

and

$$0 \leq \|\lambda'_n - \beta E_{\mu'_n, \lambda'_n} [V_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = \cdot]\|_{1, \tilde{q}_{\mu^*, \lambda^*}} \leq \left(\frac{\bar{\pi} + \bar{\iota}}{1 - \beta} + \bar{\kappa} \right) \|\tilde{q}_{\mu^*, \lambda^*} - \tilde{q}_{\mu'_n, \lambda'_n}\|_1 \\ + \|\lambda'_n - \beta E_{\mu'_n, \lambda'_n} [V_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = \cdot]\|_{1, \tilde{q}_{\mu'_n, \lambda'_n}}.$$

These facts along with (A.1) and (A.2) lets us conclude that:

$$(A.5) \quad \|V_{\mu'_n, \lambda'_n}^{\mathcal{BR}(\mu'_n, \lambda'_n)} - V_{\mu'_n, \lambda'_n}\|_{1, \hat{q}_{\mu^*, \lambda^*}} \rightarrow 0, \\ \|\lambda'_n - \beta E_{\mu'_n, \lambda'_n} [V_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = \cdot]\|_{1, \tilde{q}_{\mu^*, \lambda^*}} \rightarrow 0.$$

²⁶Under Assumption 2.1, with out loss of generality we can restrict attention to bounded exit and entry cut-off strategies. Indeed, $|\rho(x, s)|$ and $|\lambda(s)|$ are uniformly bounded by $(\bar{\pi} + \bar{\iota})/(1 - \beta) + \bar{\kappa}$.

Now since by Assumption 3.1, we must have $q_{\mu^*, \lambda^*} > 0$ component-wise, this implies that

$$(A.6) \quad V_{\mu'_n, \lambda'_n}^{\mathcal{BR}(\mu'_n, \lambda'_n)}(x, s) - V_{\mu'_n, \lambda'_n}(x, s) \rightarrow 0, \forall (x, s) \in \mathcal{Y}$$

$$(A.7) \quad \lambda'_n(s) - \beta E_{\mu'_n, \lambda'_n} [V_{\mu'_n, \lambda'_n}(x^e, s_{t+1}) | s_t = s] \rightarrow 0, \forall s \in \mathcal{S}^e.$$

Now, by Assumption 2.1 and Berge's maximum theorem, $\mathcal{BR}(\cdot)$ is continuous on $\mathcal{M} \times \Lambda$ and $V_{\mu, \lambda}^\mu$ is continuous in (μ, λ) . Thus, we have from (A.6), (A.7), and (A.3) that $V_{\mu^*, \lambda^*}^{\mathcal{BR}(\mu^*, \lambda^*)}(x, s) - V_{\mu^*, \lambda^*}(x, s) = 0, \forall (x, s) \in \mathcal{Y}$, and $\lambda^*(s) - \beta E_{\mu^*, \lambda^*} [V_{\mu^*, \lambda^*}(x^e, s_{t+1}) | s_t = s] = 0, \forall s \in \mathcal{S}^e$. Hence, $(\mu^*, \lambda^*) \in \Gamma$. But by (A.4) and the triangle inequality $d(\Gamma, (\mu^*, \lambda^*)) > \epsilon$, a contradiction. The result follows. \square

Lemma 4.1. *Let $\hat{V}_{\mu, \lambda}^*$ be an optimal solution to (4.3). We have:*

$$\|\hat{V}_{\mu, \lambda}^* - V_{\mu, \lambda}^*\|_\infty \leq \epsilon_n / (1 - \beta).$$

Proof. First, observe that (4.3) 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

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

Of course, the same upper bound must hold for $V_{\mu, \lambda}^*$. Now, since both κ and $\hat{\kappa}$ are assumed non-negative random variables, we must then have from (4.2) that

$$\|T_{\mu, \lambda}^{\text{emp}, n} \hat{V}_{\mu, \lambda}^* - T_{\mu, \lambda} \hat{V}_{\mu, \lambda}^*\|_\infty \leq \epsilon_n$$

and

$$\|T_{\mu, \lambda}^{\text{emp}, n} V_{\mu, \lambda}^* - T_{\mu, \lambda} V_{\mu, \lambda}^*\|_\infty \leq \epsilon_n,$$

so that

$$T_{\mu, \lambda} \hat{V}_{\mu, \lambda}^* \leq \hat{V}_{\mu, \lambda}^* + \epsilon_n e$$

and

$$T_{\mu, \lambda}^{\text{emp}, n} V_{\mu, \lambda}^* \leq V_{\mu, \lambda}^* + \epsilon_n e,$$

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

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

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

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

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

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

The result follows. □

Lemma 4.2. *Let $\tilde{\epsilon} < 1$ satisfy $1 - \tilde{\epsilon} \leq \frac{\mathcal{P}(x_1=x'|x_0=x, \iota_0=\lfloor \iota/\epsilon \rfloor)}{\mathcal{P}(x_1=x'|x_0=x, \iota_0=\iota)} \forall x, x', \iota$. Let $V_{\mu,\lambda}^{*,\epsilon}$ be an optimal solution to (4.5). Then:*

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

Proof. $V_{\mu,\lambda}^{*,\epsilon}$ 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}^ϵ . We show that

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

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

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

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 $\tilde{V}_{\mu,\lambda}^* = \sum_{t=0}^{\infty} \beta^t (P_{\mu,\lambda}^*)^t g^\epsilon$, and noting that the maximal absolute difference in the performance of two arbitrary strategies starting from a given state is bounded from above by $\frac{(\bar{\pi} + \bar{\iota} + \bar{\kappa})}{1-\beta}$, this lets us conclude that

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

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

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

Thus, by the triangle inequality,

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

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

$$(A.8) \quad 0 \leq V_{\mu,\lambda}^* - V_{\mu,\lambda}^{*,\epsilon} \leq \frac{\tilde{\epsilon} \beta (\bar{\pi} + \bar{\iota} + \bar{\kappa})}{(1 - \beta)^2} + \frac{d\epsilon}{1 - \beta}.$$

which yields the result. □

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

Proof. Let (r, u, t) be a feasible solution to (5.1). It is easy to see that r is a feasible solution to the LP solved in step (5) of Algorithm 3, (5.2) of the same value, for any e_j . Consequently the value of an optimal solution (r^*, u^*, t^*) to (5.1) is no larger than the value of an optimal solution to (5.2) for any e_j .

Now, (r', u', t') is by construction a feasible solution to (5.1). To see this simply note that (5.1) is

equivalent to the program

$$\begin{aligned}
& \underset{r}{\text{minimize}} && \sum_{(x,s) \in \mathcal{R}} q(x,s) \sum_{0 \leq k \leq K} \Phi_k(x,s) r_k \\
& \text{subject to} && \pi(x,s) + \mathcal{P}(\hat{\kappa} < e_j(x,s)) \left(-d\iota + \beta E_{\mu,\lambda} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right] \right) \\
& && + E[\hat{\kappa} | \hat{\kappa} \geq e_j(x,s)] \mathcal{P}(\hat{\kappa} \geq e_j(x,s)) \leq \sum_{0 \leq k \leq K} \Phi_k(x,s) r_k, \\
& && \forall (x,s) \in \mathcal{R}, \iota \in \mathcal{I} \\
& && e_j(x,s) = \max_{\iota \in \mathcal{I}} -d\iota + \beta E_{\mu,\lambda} \left[\sum_{0 \leq k \leq K} \Phi_k(x_1, s_1) r_k \middle| x_0 = x, s_0 = s, \iota_0 = \iota \right], \\
& && \forall (x,s) \in \mathcal{R}.
\end{aligned}$$

and that upon convergence in Algorithm 3,

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

for all $(x,s) \in \mathcal{R}$. Moreover, this feasible solution to (5.1) has the same value as an optimal solution to (5.2) with $e_j = t'$. Since the value of a feasible solution to (5.1) cannot exceed the value of an optimal solution to (5.2) for any value of e_j , it follows that (r', u', t') must be an optimal solution to (5.1) which proves the result. \square

B Tables

Parameters	Instances			
	No entry/exit		Entry/exit	
	High inv.	Low inv.	High inv.	Low inv.
m	75	10	100	100
θ_1	0.5	1.0	0.75	0.5
d	3.0	2.5	0.4	1.0
c	0.5	0.5	0.55	0.5
Z	1.0	0.5	1.0	1.0
\bar{x}	10	20	10	10
$\tilde{\kappa}$	–	–	250	150
$\tilde{\phi}$	–	–	100	80
x^e	–	–	3	3

Table 1: Parameter selection for comparison with MPE.

Instance	Number of firms	Long-Run Statistics							
			Total Inv.	Prod. Surp.	Cons. Surp.	C1	C2	Entry Rate	
No entry/exit	High investment	$N = 2$	MPE	0.6544	16.0598	64.9842	0.3477	0.5758	
			ALP-Based	0.6533	16.0603	64.9632	0.3484	0.5757	
			% Diff.	0.18	0.00	0.03	0.21	0.01	
		$N = 3$	MPE	0.6859	17.5846	78.7789	0.2927	0.4971	
			ALP-Based	0.6845	17.6087	78.9752	0.2924	0.4978	
		% Diff.	0.20	0.14	0.25	0.09	0.14		
	$N = 4$	MPE	0.6448	18.5182	89.0541	0.2528	0.4332		
		ALP-Based	0.6452	18.453	88.5816	0.2481	0.429		
		% Diff.	0.05	0.35	0.53	1.84	0.96		
	$N = 5$	MPE	0.5832	19.1951	97.6041	0.2211	0.3816		
		ALP-Based	0.5831	19.1344	97.0883	0.2172	0.3771		
		% Diff.	0.03	0.32	0.53	1.76	1.16		
	$N = 6$	MPE	0.4962	19.6696	104.4780	0.1936	0.3361		
		ALP-Based	0.4948	19.6337	104.1295	0.191	0.3327		
		% Diff.	0.50	0.07	0.11	0.88	0.55		
Entry/exit	Low investment	$N = 2$	MPE	0.0895	2.0509	8.2675	0.3536	0.5421	
			ALP-Based	0.0897	2.059	8.3646	0.3545	0.544	
			% Diff.	0.22	0.39	1.17	0.23	0.35	
	$N = 3$	MPE	0.0898	2.3458	10.5027	0.3263	0.493		
		ALP-Based	0.0889	2.3398	10.4654	0.3241	0.4915		
		% Diff.	1.02	0.26	0.35	0.66	0.3		
	$N = 4$	MPE	0.0754	2.5094	12.1396	0.2952	0.4485		
		ALP-Based	0.0744	2.5054	12.1083	0.2934	0.4475		
		% Diff.	1.33	0.16	0.26	0.59	0.23		
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 2: **Comparison of MPE and ALP-based indicators.** Long-run statistics computed simulating industry evolution over 10^4 periods.

Parameters	Instances			
	No entry/exit		Entry/exit	
	High inv.	Low inv.	High inv.	Low inv.
θ_1	0.3	0.5	0.75	0.5
d	0.6	0.8	0.4	1.0
c	0.5	0.5	0.55	0.5
Z	1.0	2.0	1.0	1.0
\bar{x}	20	20	15	15
\tilde{k}	-	-	250	250
$\tilde{\phi}$	-	-	40	30

Table 3: Parameter selection for comparison with OE.

Instance	Number of firms	Long-Run Statistics							
			Total Inv.	Prod. Surp.	Cons. Surp.	C6	C12	Entry Rate	
No entry/exit	High investment	$N = 20$	OE	9.2320	63.4985	550.5728	0.3531	0.6678	
			ALP-Based	9.4844	63.1420	538.5704	0.3419	0.6570	
			% Diff.	2.73	0.56	2.14	3.16	1.60	
	Low inv.	$N = 30$	OE	13.9668	95.9411	953.1851	0.2408	0.4623	
			ALP-Based	14.4277	96.5319	944.6475	0.2316	0.4512	
			% Diff.	3.30	0.61	1.60	3.81	2.39	
	$N = 50$	OE	20.65293	164.1031	1900.6840	0.1674	0.3214		
		ALP-Based	21.7132	162.7030	1745.0850	0.1619	0.3138		
		% Diff.	5.13	0.85	8.18	3.28	2.35		
Entry/exit	High 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
	Low investment	$N = 40$	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
Low 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	

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