

The Linear Programming Approach to Solving Large Scale Dynamic Stochastic Games*

Vivek Farias
MIT Sloan School

Denis Saure
Columbia Business School

Gabriel Y. Weintraub
Columbia Business School

September, 2008

Abstract

In this paper we introduce a new method to approximate Markov perfect equilibrium in large scale dynamic stochastic games 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 computed via the ‘approximate linear programming’ approach. We provide results that lend theoretical support to our approximation. We test our method on a class of dynamic models of imperfect competition. Our results suggest that the approach we propose significantly expands the set of models that can be analyzed computationally. This substantially enhances the applicability of dynamic oligopoly models among other applications of dynamic stochastic games.

1 Introduction

Dynamic stochastic games have a rich tradition in economics as a means to study strategic interactions in a changing environment (Shapley 1953). For instance, a current and important example of such a game is the Ericson and Pakes (1995) (hereafter, EP) framework for modeling 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 markets. 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

Potential applications notwithstanding, there remain substantial hurdles in the application of dynamic stochastic games as a modeling tool in practice. Dynamic stochastic games are typically analytically intractable. Therefore, solving such games entails numerically computing their Markov perfect equilibria

*Acknowledgments: The research of the first author was supported, in part, by the Solomon Buchsbaum Research Fund. The third author would like to thank Lanier Benkard and Ben Van Roy for discussions that initially stimulated this project and for useful feedback about this work. Correspondence: vivekf@mit.edu, dsau05@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).

(MPE) (e.g., Pakes and McGuire (1994)). This computation suffers from the ‘curse of dimensionality’. In a discrete-time dynamic stochastic game, each player is distinguished by an *individual state* at every point in time. For example, in an EP-type model of industrial competition the individual state captures a firm’s competitive advantage; its value could represent a measure of product quality, current productivity level, or capacity. The *system state* is a vector encoding the number of players with each possible value of the individual state variable. Assuming its competitors follow a prescribed strategy, a given player must, at each point in time, select an action to maximize its expected discounted payoffs; 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 player’s individual state and the system state; even if players were restricted to symmetric strategies, the computation entailed in selecting such an action quickly becomes infeasible as the number of players and individual states grows. For example, in a model with 30 players and 20 individual states more than one 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.

Methods that accelerate equilibrium computations have been proposed (Judd (1998), Pakes and McGuire (2001), and Doraszelski and Judd (2006)). However, computational considerations have severely limited the practical applicability of dynamic stochastic games. For example, in EP-type models computational concerns have typically limited 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, and application of the EP framework to empirical problems (the original motivation) has been rare (see Gowrisankaran and Town (1997), Benkard (2004), Jenkins, Liu, Matzkin, and McFadden (2004), Ryan (2005), Collard-Wexler (2006)).

Thus motivated, we introduce in this paper a new method to approximate MPE in large scale dynamic stochastic games. 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 stochastic games with large numbers of players, substantially enhancing the applicability of EP-type models among other applications of dynamic stochastic games.

Our method is based on an algorithm that iterates a best response operator. Stationary points of such iterations are MPE. The optimal best response value function computed at each iteration in such a scheme may be obtained via a linear program (Bertsekas 2001). Due to the curse of dimensionality, however, attempting to compute the best response at every step is computationally infeasible for many applications of interest. Indeed, the linear program that we would need to solve has an intractable number of variables and constraints. We settle instead for an approximation to the best response value function which we compute via

the ‘approximate linear programming’ approach (de Farias and Roy (2003) and de Farias and Roy (2004)). In short, the value function is approximated by a linear combination of basis functions, and 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 general dynamic stochastic games and we provide theoretical results that justify our approach. Due to their importance, we choose to numerically test our method on EP-type models. We next outline our contributions in detail.

Our first main contribution is to provide a tractable algorithm to approximate MPE in large scale dynamic stochastic games. We present an easy to follow guide to using the algorithm in general models. Among other things, we carefully address several implementation issues, such as, constraint sampling and strategy storage between iterations. 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-type 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 (2008a)). Our scheme relies on approximating the best response value function with a linear combination of basis functions. Choosing a ‘good’ approximation architecture is a problem specific task. We propose using a rich, but tractable, approximation architecture that captures a natural ‘moment’-based approximation architecture; the latter has found wide application in previous economic applications (Krusell and Smith 1998). With this approximation architecture and a suitably extended version of the approximate linear programming approach embedded in our 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 (2008b) (henceforth, OE) as a benchmark. OE is a simple to compute equilibrium concept and provides valid approximations to MPE in several EP-type models with large numbers of firms. We compare the candidate equilibrium strategy produced by our approach to OE in parameter regimes where OE can be shown to be a good approximation to MPE. Here too we show that our candidate equilibrium strategy is close to OE and hence to MPE.

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 (Weintraub, Benkard, and Van Roy 2008a). This is a commonly observed market structure in real world industries (see U.S. Economic Census). 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 problems that can be analyzed.

Our second main contribution is a series of results that give theoretical support to our approximation. These results are valid for general dynamic stochastic games. In particular, we propose a simple, easily computable convergence criterion for our algorithm that lends itself to a theoretical guarantee of the following flavor: assume that our iterative scheme converges. Further, assume that a good approximation to the value function corresponding to our candidate equilibrium strategy is within the span of our chosen basis functions. Then, upon convergence we are guaranteed to have computed a good approximation to an MPE. This result is the synthesis of several results as we now explain.

We utilize the theory developed in de Farias and Roy (2003) and de Farias and Roy (2004) that establishes that the approximate linear programming approach produces a good approximation to the best response value function provided the approximation architecture is suitably expressive. This theory requires a modest extension since the problems we must deal with involve continuous action spaces (as opposed to finite action spaces) which in turn leads us to develop and analyze a procedure that appropriately discretizes the action space. These results let us bound the magnitude by which a player 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 (Bertsekas and Tsitsiklis 1996).

Bounds of the type 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 some assumptions, 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 an MPE. The result is valid for general approximations techniques and we anticipate it can be useful to justify other approximation schemes for dynamic stochastic games or even in other contexts. Theorem 3.1 together with Theorems 4.1, 4.2, 4.3, and 4.4 provide a theoretical justification for our approximation.

As we have discussed above, our work is related to Weintraub, Benkard, and Van Roy (2008b) and Weintraub, Benkard, and Van Roy (2008a). Like them we consider algorithms that can efficiently deal with large numbers of players 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, this is the first paper that combines 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 stochastic games.

Pakes and McGuire (1994) suggested using value function approximation for EP-type 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 those developments.

Finally, our paper is related to the broader dynamic stochastic games literature that propose alternatives to dynamic programming methodologies to solve for equilibria. Breton (1991) and Filar, Schultz, Thuijssman, and Vrieze (1991) describe methods based on solving a nonlinear mathematical program. Herings and Peeters (2004) and Govindan and Wilson (2008) introduce homotopy-type algorithms. These papers solve for mixed strategy equilibria. More related to our work is Borkovsky, Doraszelski, and Kryukov (2008) that

describes a homotopy-type algorithm but focuses on pure strategy equilibrium. All these methods differ from us, however, in that their focus is on solving low dimensional stochastic games and not to overcome the curse of dimensionality.

The paper is organized as follows. In Section 2 we introduce our dynamic stochastic game 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 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 Stochastic Game

In this section we introduce a discrete-time dynamic stochastic game with a finite number of states (Shapley (1953), Filar and Vrieze (1997), and Basar and Olsder (1999)). In Section 2.1 we introduce the model and notation. In Section 2.2 we introduce the notion of Markov perfect equilibrium. Finally, in Section 2.3, we specialize the dynamic stochastic game model to an EP-type model.

2.1 Model and Notation

We consider an infinite horizon discrete-time stochastic game. We index time periods with nonnegative 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.

There are N players indexed by $\mathcal{I} = \{1, \dots, N\}$. Players' heterogeneity is reflected through their *individual states*. At time t , the individual state of player $i \in \mathcal{I}$ is denoted by $x_{i,t} \in \mathcal{X}$, where \mathcal{X} is a finite set. We define the *system state* s_t to be a vector over individual states that specifies, for each $x \in \mathcal{X}$, the number of players at state x in period t . For each $i \in \mathcal{I}$, we define $s_{-i,t}$ to be the state of the *competitors* of player i ; that is, $s_{-i,t}(x) = s_t(x) - 1$ if $x_{i,t} = x$, and $s_{-i,t}(x) = s_t(x)$, otherwise. We will use the notation (x, s) to refer to a player in state x with competitors in state s . Finally, we define the state space $\mathcal{S} = \left\{s \in \mathbb{N}^{|\mathcal{X}|} \mid \sum_{x \in \mathcal{X}} s(x) = N - 1\right\}$.

In each period, each agent takes an action $a \in \mathcal{A}$, where \mathcal{A} is a closed interval of the real line $[a, \bar{a}]$.² We assume the state evolution for a player depends on its own current state and the action it takes. Formally, if

²It is straightforward to extend the model and assume that \mathcal{A} is a convex and compact subset of an Euclidian space.

player i takes action $a_{i,t}$ at time period t , then its state at time period $t + 1$ is given by

$$x_{i,t+1} = f(x_{i,t}, a_{i,t}, \zeta_{i,t+1}),$$

where the random variables $\{\zeta_{i,t}|t \geq 0, i \in \mathcal{I}\}$ are independent and identically distributed, and reflect idiosyncratic uncertainty in the state evolution.

In each period, each agent receives a payoff. A player's single-period expected payoff $\pi(x_{i,t}, s_{-i,t}, a_{i,t})$ depends on its individual state $x_{i,t} \in \mathcal{X}$, its competitors' state $s_{-i,t} \in \mathcal{S}$, and its action $a_{i,t} \in \mathcal{A}$.

Each player aims to maximize expected net present value. We assume a constant discount factor of $\beta \in (0, 1)$ per time period.

2.2 Markov Perfect Equilibrium

As is common in many applications of dynamic stochastic games, we focus on pure strategy Markov perfect equilibrium (MPE) (see Maskin and Tirole (1988) and Ericson and Pakes (1995) for significant examples in economics). We further assume that equilibrium is symmetric, such that all players use a common stationary strategy. In particular, there is a function μ such that at each time t , each player $i \in \mathcal{I}$ plays action $a_{i,t} = \mu(x_{i,t}, s_{-i,t})$. Let \mathcal{M} denote the set of strategies such that an element $\mu \in \mathcal{M}$ is a function $\mu : \mathcal{X} \times \mathcal{S} \rightarrow \mathcal{A}$.

We define the value function $V_{\mu}^{\mu'}(x, s)$ to be the expected net present value for a player at state x when its competitors' state is s , given that its competitors each follows a common strategy $\mu \in \mathcal{M}$, and the agent itself follows strategy $\mu' \in \mathcal{M}$. In particular,

$$V_{\mu}^{\mu'}(x, s) = E_{\mu}^{\mu'} \left[\sum_{k=0}^{\infty} \beta^k \pi(x_{i,k}, s_{-i,k}, a_{i,k}) \middle| x_{i,0} = x, s_{-i,0} = s \right],$$

where i is taken to be the index of a player at state x at time 0 and the superscript and subscripts of the expectation indicate the strategy followed by player i , and the strategy followed by its competitors. In an abuse of notation, we will use the shorthand, $V_{\mu}(x, s) \equiv V_{\mu}^{\mu}(x, s)$, to refer to the expected discounted value of payoffs when player i follows the same strategy μ as its competitors.

A MPE to our model comprises a strategy $\mu \in \mathcal{M}$ that satisfies:

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

Standard dynamic programming arguments establish that the supremum above can always be attained simultaneously for all x and s by a common strategy μ' . We introduce the following additional assumption

that holds throughout the paper.

Assumption 2.1.

1. *There exists $\bar{\pi} < \infty$, such that, $|\pi(x, s, a)| \leq \bar{\pi}$, for all $x \in \mathcal{X}$, $s \in \mathcal{S}$, $a \in \mathcal{A}$.*
2. *For all $x \in \mathcal{X}$, $s \in \mathcal{S}$, $\pi(x, s, a)$ is continuous in a .*
3. *For all $x, x' \in \mathcal{X}$, $\mathcal{P}[f(x', a, \zeta_{i,t+1}) = x]$ is continuous in a .*
4. *For all $\mu \in \mathcal{M}$, $V \in \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$, and $(x, s) \in \mathcal{X} \times \mathcal{S}$, the problem $\max_{a \in \mathcal{A}} \pi(x, s, a) + \beta E_{\mu}[V(x_{i,1}, s_{-i,1}) | x_{i,0} = x, s_{-i,0} = s, a_{i,0} = a]$ admits a unique optimal solution.*
5. *For all strategies $\mu \in \mathcal{M}$, the Markov chain that describes the individual state evolution $\{x_{i,t} : t \geq 0\}$ is irreducible and aperiodic.*

Previous work establishes that under Assumptions 2.1.1-2.1.4 an MPE always exists (Doraszelski and Satterthwaite (2007) and Escobar (2006)). In particular, Assumption 2.1.4 ensures a unique solution to the players' decision problem. The assumption is used to guarantee existence of an equilibrium in pure strategies. Similar assumptions are common in the literature. For example, Assumption 2.1.4 is satisfied by most EP-type models analyzed in previous work (Doraszelski and Satterthwaite 2007). With respect to uniqueness, in general we presume that our model may have multiple equilibria. Indeed, Doraszelski and Satterthwaite (2007) provide an example of multiple equilibria for an economic model similar to the one we introduce below in Section 6. Assumption 2.1.5 together with the fact that the state space is finite, imply that the Markov chain that describes the state evolution $\{s_t : t \geq 0\}$ admits a unique invariant distribution. While weaker assumptions can be made to ensure that, Assumption 2.1.5 is useful to prove Theorem 3.1. The model we introduce in Section 6 satisfies Assumption 2.1.

2.3 Dynamic Oligopoly Model

In this section we explain how the dynamic stochastic game above can be specialized to model dynamic competition in an oligopolistic industry like Ericson and Pakes (1995). Motivated by the fact that a blossoming recent literature has applied EP-type models to numerous applied problems in economics concerning dynamic competition (see Doraszelski and Pakes (2007)), we will test our model in a class of EP models (see Section 6 for more details).

In a typical EP-model players represent firms competing in an industry. An individual firm state may reflect its quality level, productivity, capacity, the size of its consumer network, or any other aspect of the firm that affects its profits. To fix an interpretation in the context of an EP-model, we will refer to a firm's state as its quality level and we will define $\mathcal{X} = \{0, \dots, \bar{x}\}$. The integer number \bar{x} is an upper bound on firms' quality levels.

An action represents an investment to improve the quality level. If a firm invests $a_{i,t} \in [0, \bar{a}]$, then the firm's state at time $t + 1$ is given by,

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

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. Note that this specification is very general as w may take on either positive or negative values (e.g., allowing for positive depreciation). There is a unit cost of investment denoted by d .

In each period, each incumbent firm earns profits on a spot market. A firm's single period expected profit $\pi(x_{i,t}, s_{-i,t})$ depends on its quality level $x_{i,t} \in \mathcal{X}$ and its competitors' state $s_{-i,t} \in \mathcal{S}$. 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. In this model, $\pi(x, s)$ is typically increasing in x so that higher quality levels imply larger profits. Also, $w(a, \zeta)$ is increasing in a so that investment is productive.

EP-type models also allow for the entry and exit of firms. In each period, there could be a set of potential entrants that may decide to enter the industry after paying a setup cost. In each period, incumbent firms may choose to exit the industry, earn a sell-off value and cease operations permanently. The model can also accommodate industry-wide shocks that affect all firms equally. The model with entry and exit decisions, and aggregate shocks is also a dynamic stochastic game similar to the one introduced above. In the computational experiments that we present in Section 6, however, we consider an EP-type model with a fixed number of firms without entry and exit decisions. We also assume that all shocks are idiosyncratic. However, as we argue in the conclusions, we believe that our method can be easily extended to accommodate these more general models.

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. That motivates our approach of finding *approximate* best responses at every step instead. We provide a theoretical justification for our approach in Theorem 3.1, where we show that as we improve the accuracy of the approximate best responses, we get

closer to an 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 players' strategies at each step. Stationary points of such iterations are MPE. While in many cases, including ours, the theoretical convergence properties of best response algorithms are not well understood (Fudenberg and Levine 1998), positive practical experience support their use.³

With this motivation, for all $\mu \in \mathcal{M}$, we define the best response operator $F : \mathcal{M} \rightarrow \mathcal{M}$ according to

$$F(\mu) = \mu^*, \text{ where } \sup_{\mu' \in \mathcal{M}} V_{\mu'}^{\mu'}(x, s) = V_{\mu^*}^{\mu^*}(x, s), \quad \forall x \in \mathcal{X}, \forall s \in \mathcal{S}.$$

A fixed point of the operator F is an MPE. We introduce the following algorithm:

Algorithm 1 Best Response Algorithm for MPE

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

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

Step (4) in the algorithm requires solving a dynamic programming problem to optimize players' strategies. The size of the state space of this problem is equal to:

$$|\mathcal{X}| \binom{N + |\mathcal{X}| - 2}{N - 1}.$$

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 players and 20 individual states has hundreds of billions of states. This motivates our alternative approach which relaxes the requirement of finding a best response in step (4) of the algorithm. To formalize this idea we introduce the following definitions.

Definition 3.1. Let $V_{\mu}^{\mu^*} = \sup_{\mu' \in \mathcal{M}} V_{\mu'}^{\mu'}$. Given $\tilde{\mu} \in \mathcal{M}$, let $q_{\tilde{\mu}}^{\tilde{\mu}}$ be the invariant distribution of the Markov chain $\{(x_{i,t}, s_{-i,t}) : t \geq 0\}$ induced over $\mathcal{X} \times \mathcal{S}$ by a player using strategy $\tilde{\mu}$ in response to the competitor

³Doraszelski and Pakes (2007) discuss other dynamic programming based methods to compute MPE that have been used in previous studies. There is no guarantee of convergence for these methods either.

strategy μ . We call $\tilde{\mu}$ an ϵ -weighted best response to $\mu \in \mathcal{M}$ if⁴

$$\|V_{\mu}^{\mu^*} - V_{\mu}^{\tilde{\mu}}\|_{1, q_{\mu}^{\tilde{\mu}}} \leq \epsilon.$$

Definition 3.2. $\tilde{\mu} \in \mathcal{M}$ is an ϵ -weighted MPE if $\tilde{\mu}$ is an ϵ -weighted best response to itself.

Under our definition, the maximum potential gain to a player in deviating from an ϵ -weighted MPE, $\tilde{\mu}$, is averaged across states, with an emphasis on states visited frequently under $\tilde{\mu}$; this average gain can be at most ϵ . When resorting to our approximation it will be unlikely that one can approximate the value function 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 is similar to other concepts that have been previously used to assess the accuracy of approximations to MPE and as stopping criteria (see Weintraub, Benkard, and Van Roy (2008b) and Pakes and McGuire (2001)).

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 $\mu_{i+1} := \tilde{F}(\mu_i)$ for which we will have that

$$\begin{aligned} (3.1) \quad \|V_{\mu_i}^{\mu^*} - V_{\mu_i}^{\mu_i}\|_{1, q_{\mu_i}^{\mu_i}} &\leq \|V_{\mu_i}^{\mu^*} - V_{\mu_i}^{\mu_i}\|_{1, q_{\mu_i}^{\mu_{i+1}}} + \frac{\bar{\pi}}{1 - \beta} \|q_{\mu_i}^{\mu_{i+1}} - q_{\mu_i}^{\mu_i}\|_1 \\ &\leq \|V_{\mu_i}^{\mu^*} - V_{\mu_i}^{\mu_{i+1}}\|_{1, q_{\mu_i}^{\mu_{i+1}}} + \|V_{\mu_i}^{\mu_{i+1}} - V_{\mu_i}^{\mu_i}\|_{1, q_{\mu_i}^{\mu_{i+1}}} + \frac{\bar{\pi}}{1 - \beta} \|q_{\mu_i}^{\mu_{i+1}} - q_{\mu_i}^{\mu_i}\|_1. \end{aligned}$$

Now if \tilde{F} were guaranteed to compute an $\epsilon/2$ -best response, we would have that $\|V_{\mu_i}^{\mu^*} - V_{\mu_i}^{\mu_{i+1}}\|_{1, q_{\mu_i}^{\mu_{i+1}}} < \epsilon/2$. Moreover the last two terms on the right hand side of (3.1) can be estimated without knowledge of μ^* so that if we select as a stopping criterion the requirement that $\|V_{\mu_i}^{\mu_{i+1}} - V_{\mu_i}^{\mu_i}\|_{1, q_{\mu_i}^{\mu_{i+1}}} + \frac{\bar{\pi}}{1 - \beta} \|q_{\mu_i}^{\mu_{i+1}} - q_{\mu_i}^{\mu_i}\|_1$ be sufficiently small (say, $< \epsilon/2$), then with the fact that \tilde{F} computes an $\epsilon/2$ -best response, we would have that upon convergence, the algorithm computes an ϵ -weighted MPE.

Notice that our ability to compute ϵ -weighted MPE relies crucially on the quality of the approximation we can provide to the best response at each stage of the algorithm, that is, the magnitude of $\|V_{\mu_i}^{\mu^*} - V_{\mu_i}^{\mu_{i+1}}\|_{1, q_{\mu_i}^{\mu_{i+1}}}$. Our proposed method is motivated by the fact that as we improve our approximation to the best response we are able to find ϵ -weighted MPE with smaller values of ϵ . As ϵ converges to zero, our approach produces a strategy that permits vanishingly small gains to deviating in states that have positive probability of occurrence under the invariant distribution. It is natural then to expect that the strategy

⁴For $c \in \mathbb{R}_+^k$, the $(1, c)$ norm of a vector $x \in \mathbb{R}^k$ is defined according to $\|x\|_{1, c} = \sum_{i=1}^k |x_i| c_i$.

we converge to approaches an MPE in relevant states. In Section 6 we present computational experiments that support this point. Moreover, under Assumption 2.1 we can prove that the strategies we converge to indeed approach an MPE. In what follows, let $\Gamma \subseteq \mathcal{M}$ be the set of MPE. For all $\mu \in \mathcal{M}$, let us define $d(\Gamma, \mu) = \inf_{\mu' \in \Gamma} \|\mu' - \mu\|_\infty$. We have the following theorem whose proof may be found in the appendix.

Theorem 3.1. *Let $\{\mu_n \in \mathcal{M} | n \in \mathbb{N}\}$ be a sequence of ϵ_n -weighted MPE. Suppose that $\lim_{n \rightarrow \infty} \epsilon_n = 0$. Then, $\lim_{n \rightarrow \infty} d(\Gamma, \mu_n) = 0$.*

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}^Γ 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, the best response is found by solving a linear program for which the number of variables is equal to the size of the state space and the number of constraints is equal to the size of the state space times the size of the action space.

Following de Farias and Roy (2003) and de Farias and 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 linear program substantially from the size of the state space to the number of basis functions. However, the number of constraints is still prohibitive. In section 4.3 we describe how discretization of the action space together with 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 an MPE strategy in the sense of Theorem 3.1.

4.1 Linear Programming Approach to MPE

For a fixed $\mu \in \mathcal{M}$, consider the problem of computing a best response strategy μ_μ^* . To this end, let us define for every strategy $\mu' \in \mathcal{M}$, an operator $T_\mu^{\mu'} : \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$ according to

$$(T_\mu^{\mu'} V)(x, s) = \pi(x, s, \mu'(x, s)) + \beta E_\mu[V(x_1, s_1) | x_0 = x, s_0 = s, a_0 = \mu'(x, s)], \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S},$$

where $V \in \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$. To simplify notation, we omit the dependence on i and $-i$ in $x_{i,t}$ and $s_{-i,t}$ whenever it does not lead to ambiguities.

We define the Bellman operator $T_\mu : \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$ according to

$$T_\mu V = \sup_{\mu' \in \mathcal{M}} T_\mu^{\mu'} V,$$

where the supremum is attained componentwise. A best response strategy to μ , which we denote with a minor abuse of notation by μ^* , may be found by first computing a fixed point of the Bellman operator. In particular, define V_μ^* as the unique solution to Bellman's equation

$$(4.1) \quad T_\mu V = V.$$

It is simple to show that $V_\mu^* = \sup_{\mu' \in \mathcal{M}} V_\mu^{\mu'} = V_\mu^{\mu^*}$. A best response strategy may then be found as the greedy maximizer with respect to V_μ^* in Bellman's equation (Bertsekas 2001). That is, a best response strategy μ^* may be identified as a strategy for which

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

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 (Bertsekas 2001).

$$(4.2) \quad \begin{aligned} & \min \quad c'V \\ & \text{s.t.} \quad (T_\mu V)(x, s) \leq V(x, s), \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S}. \end{aligned}$$

Provided that $c \in \mathbb{R}_+^{|\mathcal{X} \times \mathcal{S}|}$ is component-wise positive, V_μ^* is the unique optimal solution to the above program. The program is non-linear since the constraints are non-linear. However, the program can be

rewritten as the following semi-infinite linear program:

$$\begin{aligned}
(4.3) \quad & \min \quad c'V \\
& \text{s.t.} \quad \pi(x, s, a) + \beta E_\mu[V(x_1, s_1) | x_0 = x, s_0 = s, a_0 = a] \leq V(x, s), \\
& \quad \quad \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S}, \forall a \in \mathcal{A}.
\end{aligned}$$

The above program is intractable. In particular, it has an uncountable number of constraints due to the fact that the number of actions one may consider at any state is uncountable. Even if this were not the case, that is, restricting attention to a finite number of actions at each state, one still has to contend with the curse of dimensionality. In particular, the relevant program has $|\mathcal{X} \times \mathcal{S}|$ variables and at least that many constraints; as we have discussed $|\mathcal{X} \times \mathcal{S}|$ 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. First, in Section 4.2 we develop a linear program with a small number of variables that computes a good approximation to the optimal value function. This linear program will have a small number of variables but a large number of constraints. That section will assume that we can solve this program. In section 4.3 we address the large number of constraints we must contend with via an appropriate discretization of the action space and a constraint sampling scheme.

4.2 Value Function Approximation

Assume we are given a set of “basis” functions $\phi_i : \mathcal{X} \times \mathcal{S} \rightarrow \mathbb{R}$, for $i = 1, 2, \dots, k$. Let us denote by $\Phi \in \mathbb{R}^{|\mathcal{X} \times \mathcal{S}| \times k}$ the matrix $[\phi_1, \phi_2, \dots, \phi_k]$. Given the difficulty in computing V_μ^* exactly, we focus in this section on computing a set of weights $r \in \mathbb{R}^k$ for which Φr closely approximates V_μ^* . To that end, we consider the following program:

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

Similarly to (4.3), the above program can be rewritten as a semi-infinite linear program. The above program attempts to find a good approximation to V_μ^* 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_μ^* , then the program (4.4) should provide an effective approximation. The following Theorem proved in de Farias and Roy (2003) formalizes this notion.

Theorem 4.1. *Let e , the vector of ones, be in the span of the columns of Φ and c be a probability distribution.*

Let r_μ be an optimal solution to (4.4). Then,

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

By settling for an approximation to the optimal value function, we have reduced our problem to the solution of a linear program with a potentially small number of variables (k). The number of constraints that we must contend with continues to remain large, and we will eventually resort to a constraints sampling scheme together with action space discretization to compute a solution to this program. Nonetheless, (4.4) represents a substantial simplification to the program (4.2) we started with.

Given a good approximation to V_μ^* , namely Φr_μ one may consider using as a proxy for the best response strategy the greedy strategy with respect to Φr_μ , namely, a strategy $\tilde{\mu}$ satisfying

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

Provided Φr_μ is a good approximation to V_μ^* , the expected discounted payoffs associated with using strategy $\tilde{\mu}$ in response to competitors that use strategy μ is also close to V_μ^* as is made precise by the following result which is easy to establish (see de Farias and Roy (2003)).

Theorem 4.2. *Let $q_\mu^{\tilde{\mu}}$ be the invariant distribution of the Markov chain $\{(x_{i,t}, s_{-i,t}) : t \geq 0\}$ induced over states in $\mathcal{X} \times \mathcal{S}$ by a player using strategy $\tilde{\mu}$ in response to μ . Then,*

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

Together with Theorem 4.1, this result lets us conclude that

$$(4.5) \quad \|V_\mu^{\tilde{\mu}} - V_\mu^*\|_{1,q_\mu^{\tilde{\mu}}} \leq \max_{x,s} \frac{q_\mu^{\tilde{\mu}}(x,s)}{c(x,s)} \frac{2}{(1-\beta)^2} \inf_r \|\Phi r - V_\mu^*\|_\infty.$$

It is worth pausing to discuss what we have established thus far. Assume we could solve the program (4.4) and thus compute an approximate best response $\tilde{\mu}$ at every stage of Algorithm 1. $\tilde{\mu}$ would then be an ϵ -weighted best response with ϵ specified by the right hand side of (4.5). In particular, assume that upon convergence our iterative best response scheme converged to some strategy $\bar{\mu}$. Let μ^* be an approximate best response to $\bar{\mu}$. Then, by (3.1), $\bar{\mu}$ would constitute an ϵ -weighted MPE with

$$(4.6) \quad \epsilon = \max_{x,s} \frac{q_\mu^{\mu^*}(x,s)}{c(x,s)} \frac{2}{(1-\beta)^2} \inf_r \|\Phi r - V_\mu^*\|_\infty + \epsilon',$$

where ϵ' can be made arbitrarily small by selecting an appropriate stopping criterion. 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 be 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 (4.4) is equivalent to the program (see de Farias and Roy (2003)):

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

As suggested by (4.6), these state relevance weights should ideally be close to the invariant distribution induced over states by an approximate best response to the candidate equilibrium strategy.

4.3 Reducing the Number of Constraints

The previous section reduced the problem of finding an approximate best response to a given strategy μ to the solution of a semi-infinite linear program with a small number of variables. This section focuses on developing a practical scheme to approximately solve such a program. In particular, we resort to a two step procedure to simplify this program by reducing the number of constraints. In particular, we first show that restricting attention to constraints corresponding to some finite subset of actions in \mathcal{A} suffices to produce a good approximation to the optimal solution of (4.4). This is tantamount to a discretization of the set of potential actions for a player and yields a program with a small number of variables and a large but finite number of constraints whose solution is close to the optimal solution to (4.4). While finite, this program will have a number of constraints no smaller than $|\mathcal{X} \times \mathcal{S}|$, which is far too large a number for many problems of interest. As such we will use a constraint sampling procedure to further simplify the program we must solve.

We begin by first discretizing the set of permissible actions each player is allowed to make in responding to its competitors' strategy μ in any state. In particular, let us define for arbitrary $\epsilon > 0$, the set $\mathcal{A}^\epsilon = \{\underline{a}, \underline{a} + \epsilon, \underline{a} + 2\epsilon, \dots, \underline{a} + \lfloor (\bar{a} - \underline{a})/\epsilon \rfloor \epsilon\}$ and with a minor abuse of notation define a "discretized" Bellman

operator $T_\mu^\epsilon : \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$ according to

$$(4.7) \quad (T_\mu^\epsilon V)(x, s) = \max_{\mu'(x, s) \in \mathcal{A}^\epsilon} (T_\mu^{\mu'} V)(x, s), \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S}.$$

Let us denote by $V_\mu^{*, \epsilon}$ the value function corresponding to a best response strategy to μ when actions are restricted to the set \mathcal{A}^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 $V_\mu^{*, \epsilon}$. The value function $V_\mu^{*, \epsilon}$ is the unique fixed point of the discretized Bellman operator T_μ^ϵ . We introduce the discretization via the following linear programming relaxation of the semi-infinite linear program (4.4):

$$(4.8) \quad \begin{aligned} \min \quad & c' \Phi r \\ \text{s.t.} \quad & (T_\mu^\epsilon \Phi r)(x, s) \leq (\Phi r)(x, s) \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S}. \end{aligned}$$

The relaxed program (4.8) can be rewritten as a standard linear program with k variables and $|\mathcal{X} \times \mathcal{S} \times \mathcal{A}^\epsilon|$ constraints. Provided one chooses a sufficiently fine discretization of the set of actions, this program is also capable of producing a good approximation to V_μ^* , that is, an approximation whose quality scales gracefully with $\inf_r \|\Phi r - V_\mu^*\|_\infty$. In particular, we have the following theorem whose proof may be found in the appendix.

Theorem 4.3. *Let $\tilde{\epsilon} < 1$ satisfy $1 - \tilde{\epsilon} \leq \frac{\mathcal{P}(f(x, a + |(a-a)/\epsilon| \epsilon, \zeta) = x')}{\mathcal{P}(f(x, a, \zeta) = x')}$, $\forall x, x', a$. Moreover, let $\pi(x, s, \cdot)$ have Lipschitz constant K for all x, s .⁵ Finally, let r_μ^ϵ be an optimal solution to (4.8). Then:*

$$\|\Phi r_\mu^\epsilon - V_\mu^*\|_{1,c} \leq \frac{2}{1 - \beta} \inf_r \|\Phi r - V_\mu^*\|_\infty + \frac{3 - \beta}{1 - \beta} \left(\frac{2\tilde{\epsilon}\beta\pi}{(1 - \beta)^2} + \frac{K\epsilon}{1 - \beta} \right).$$

The linear program (4.8) we have developed has a small number of variables but a potentially enormous number of constraints making exact solution difficult. Instead, we will settle for a near feasible solution whose value is close to that of an optimal solution. We will compute this solution by sampling a tractable number of constraints from the set of constraints of the LP (4.8) and solving a program with an objective identical to (4.8) but only with the sampled constraints. In particular, given an arbitrary sampling distribution over states in $\mathcal{X} \times \mathcal{S}$, ψ , one may consider sampling a set \mathcal{R} of L states in $\mathcal{X} \times \mathcal{S}$ according to ψ . We consider solving the following relaxation of (4.8):

$$(4.9) \quad \begin{aligned} \min \quad & c' \Phi r \\ \text{s.t.} \quad & (T_\mu^\epsilon \Phi r)(x, s) \leq (\Phi r)(x, s) \quad \forall (x, s) \in \mathcal{R}. \end{aligned}$$

⁵Note that the function $\pi(x, s, \cdot)$ is continuous over a compact set, hence it is Lipschitz continuous.

Intuitively, a sufficiently large number of samples L should suffice to guarantee that a solution to (4.9) satisfies all except a small set of constraints in (4.8) with high probability. In fact, it has been shown in de Farias and 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 $\mu^{*,\epsilon}$ and define ψ^* according to

$$\psi^*(x, s) = (1 - \beta)E_{\mu}^{\mu^{*,\epsilon}} \left[\sum_{t=0}^{\infty} \beta^t \mathbf{1}\{x_{i,t} = x, s_{-i,t} = s\} | (x_{i,0}, s_{-i,0}) \sim c \right].$$

ψ^* gives the expected discounted number of visits to a particular state when the strategy $\mu^{*,\epsilon}$ is used in response to μ and the initial state is sampled according to c . We do not have access to ψ^* ; let $\bar{\psi}$ be a sampling distribution satisfying $\max_{x,s} \frac{\psi^*(x,s)}{\bar{\psi}(x,s)} \leq C$. Assuming the L states in \mathcal{R} are sampled according to $\bar{\psi}$, we then have the following result, specialized from de Farias and Roy (2004):

Theorem 4.4. *Let $\delta, \epsilon' \in (0, 1)$. Let \mathcal{R} consist of L states in $\mathcal{X} \times \mathcal{S}$ sampled according to $\bar{\psi}$. Let $\tilde{r}_{\mu}^{\epsilon}$ be an optimal solution to (4.9). If*

$$L \geq \frac{16\|V_{\mu}^{*,\epsilon} - \Phi\tilde{r}_{\mu}^{\epsilon}\|_{\infty}C}{(1 - \beta)\epsilon'c^T V_{\mu}^{*,\epsilon}} \left(K \ln \frac{48\|V_{\mu}^{*,\epsilon} - \Phi\tilde{r}_{\mu}^{\epsilon}\|_{\infty}C}{(1 - \beta)\epsilon'c^T V_{\mu}^{*,\epsilon}} + \ln \frac{2}{\delta} \right),$$

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

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

The result and the discussion in de Farias and Roy (2004) suggest that sampling a tractable number of constraints according to a distribution close to ψ^* ensures that $\|V_{\mu}^{*,\epsilon} - \Phi\tilde{r}_{\mu}^{\epsilon}\|_{1,c} \approx \|V_{\mu}^{*,\epsilon} - \Phi r_{\mu}^{\epsilon}\|_{1,c}$. Of course, we do not have access to ψ^* ; ψ^* requires we already have access to a best response to μ . Nonetheless, our sequential MPE computation yields a natural candidate for $\bar{\psi}$: in every iteration we simply sample states according to the approximate best response strategy computed at the prior iteration.

Theorems 4.3 and 4.4 together establish the quality of our approximation to a best response computed via a *tractable* linear program (4.9). In particular, we showed that by choosing a suitable discretization of $[\underline{a}, \bar{a}]$ and 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 (4.4); 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 $F(\mu)$ by a strategy $\tilde{\mu}$ satisfying $T_{\mu}^{\tilde{\mu}} \Phi \tilde{r}_{\mu}^{\epsilon} = T_{\mu} \Phi \tilde{r}_{\mu}^{\epsilon}$ where $\tilde{r}_{\mu}^{\epsilon}$ is an optimal solution to the tractable LP (4.9). A number of details, such as the choice of basis functions Φ and the sampling distribution ψ^* , remain unresolved. These will be addressed in subsequent sections where we describe our algorithm to compute an approximation to MPE precisely and discuss our computational experiments.

5 Algorithm

This section specifies an implementable algorithm for approximate MPE computation using the machinery developed in the previous section. This requires specifying several details of the algorithm. In particular, in Section 5.1 we discuss the constraint sampling distribution we use. In Section 5.2 we introduce our algorithm to approximate MPE. We also address issues that arise when an iterative best response scheme of the form of Algorithm 1 is used in conjunction with an approximation algorithm of the type discussed in the previous section. Specifically, we address issues related to strategy storage which are discussed in Section 5.3.

5.1 Constraint Sampling Distribution and State-Relevance Weights: ψ and c

The previous section developed a linear program, (4.9), to compute an approximate best response to strategy μ . This requires an appropriate selection of state-relevance weights, c , and a constraint sampling distribution, $\bar{\psi}$. There is little understanding in the literature of what might constitute a computable “optimal” choice of c . A common heuristic is to take c to be the invariant distribution induced by a good approximation to the optimal strategy. This choice is lent some support by the observation in Section 4.2 that c effectively trades-off performance loss across states and the expression for performance loss (4.5), which suggests that we select c to be as close to the invariant distribution induced by a best response. With this motivation we take

$$c(x, s) = \hat{c}_T^{\mu}(x, s) = \frac{1}{T} \left[\sum_{t=0}^T \mathbf{1}\{x_t = x, s_t = s\} | (x_0, s_0) \sim \nu \right],$$

for some suitable large T , where $\{(x_t, s_t)\}$ is a sample trajectory of states visited assuming all players employ strategy μ and the initial state is sampled according to a distribution ν .

Theorem 4.4 suggests that we use the idealized distribution ψ^* to sample states from $\mathcal{X} \times \mathcal{S}$. Of course, since we do not have access to $\mu^{*,\epsilon}$ (this is what we are after in the first place), we settle for the following

alternative:

$$\hat{\psi}_T^\mu(x, s) = \hat{c}_T^\mu(x, s).$$

This sampling distribution is likely to be a good approximation to ψ^* , provided T and β are large, and μ is a good approximation to $\mu^{*,\epsilon}$. In the iterative method to compute MPE that we introduce below, we envision that as we get closer to an MPE the best response strategy computed at the previous iteration, μ , will provide a good approximation to $\mu^{*,\epsilon}$.⁶

We are now ready to provide a precise specification of a subroutine we will use to compute an approximation to the best response to a given strategy μ . The subroutine will assume the existence of an oracle to compute $\mu(x, s)$; we will specify that oracle in the context of our iterative MPE computation scheme later. The subroutine takes as input the following parameters: L , the number of sampled states; ϵ , the tolerance parameter for discretization of the action set; and ν , an arbitrary distribution over states. Given a strategy μ for competitors, Algorithm 2 computes an approximation to the best response value function $V_\mu^{\mu^*}$.

Algorithm 2 $\tilde{G}(\mu, L, \epsilon, \nu)$ (Subroutine to Compute Approximate Value Function)

- 1: Select $(x_0, s_0) \sim \nu$.
- 2: Sample L states (x, s) according to distribution $\hat{\psi}_T^\mu$ to generate set \mathcal{R} .
- 3: Solve

$$\begin{aligned} \min \quad & \sum_{(x,s) \in \mathcal{R}} (\Phi r)(x, s) \\ \text{s.t.} \quad & (T_\mu^\epsilon \Phi r)(x, s) \leq (\Phi r)(x, s) \quad \forall (x, s) \in \mathcal{R} \end{aligned}$$

- 4: Output optimal solution to above program, \tilde{r}_μ^ϵ .
-

Given \tilde{r}_μ^ϵ , an approximate best response to μ , $\tilde{F}(\mu, \tilde{r}_\mu^\epsilon)$, is computed according to

$$\tilde{F}(\mu, \tilde{r}_\mu^\epsilon)(x, s) \in \operatorname{argmax}_{a \in \mathcal{A}^\epsilon} \pi(x, s, a) + \beta E_\mu[(\Phi \tilde{r}_\mu^\epsilon)(x_1, s_1) | x_0 = x, s_0 = s, a_0 = a].$$

Before moving on to integrate the above subroutine into our scheme for approximate MPE computation, several remarks are in order.

Remark 5.1. Notice that expressing a constraint in the linear program in the above subroutine requires we compute expectations of the form $E_\mu[V(x_1, s_1) | x_0 = x, s_0 = s, a_0 = a]$, assuming one has access to an oracle that computes $\mu(x, s)$. Under the model and separable approximation architecture we introduce in the next section, this computation is simplified and requires approximately $\Theta(|\mathcal{X}|N^4)$ operations.⁷

⁶These choices for c and ψ may lead to the unboundedness of program (4.9). In particular, variables that appear in the objective function may remain unconstrained in the sampled feasible region. To avoid such situations, we require that each coefficient r in the approximation Φr is bounded below by a constant. Our numerical experiments show that such a modification does not impact the quality of the solution to (4.9), provided enough constraints are sampled.

⁷In that model, players can only transition to adjacent individual states. Considering this and the separable nature of the

Remark 5.2. *In general, the strategy we must best respond to, μ , cannot be stored in a look up table since such a table would have size $\Theta(|\mathcal{X} \times \mathcal{S}|)$. Instead, we will have that μ is specified as the greedy strategy with respect to some second strategy μ_{-1} and an approximation to the value function specified by weights r_{-1} , i.e.,*

$$\mu(x, s) \in \operatorname{argmax}_{a \in \mathcal{A}^\epsilon} \pi(x, s, a) + \beta E_{\mu_{-1}}[(\Phi r_{-1})(x_1, s_1) | x_0 = x, s_0 = s, a_0 = a].$$

μ_{-1} in turn may either be specified in a similar fashion as above (with weights r_{-2} and some strategy μ_{-2}), or else have some implicit compact representation (such as say, the greedy one-period payoff maximizing strategy). Our oracle to compute μ , M , will thus in general take as input a sequence of weight vectors $r_{-1}, r_{-2}, \dots, r_{-l}$ and a strategy with a compact representation μ_{-l} . We specify the oracle in Section 5.3.

5.2 Algorithm for Approximating MPE

We now present our overall scheme for computing an approximation to MPE that uses the above subroutine and assumes an efficient oracle M to generate a strategy given a sequence of weight vectors $r_{-1}, r_{-2}, \dots, r_{-l}$ and a strategy with a compact representation μ_{-l} . Algorithm 3 takes as input the specification of the model and what are essentially ‘tuning’ parameters L, ϵ and ν (required for the approximate best response subroutine viz. Algorithm 2 above), along with an initial strategy with a compact representation. An example of such a strategy is the greedy strategy μ^g defined according to:

$$\mu^g(x, s) \in \operatorname{argmax}_{a \in \mathcal{A}} \pi(x, s, a).$$

As suggested in Section 3, we select as a stopping criterion

$$\Delta = \|V_{\mu_i}^{\mu_{i+1}} - V_{\mu_i}^{\mu_i}\|_{1, q_{\mu_i}^{\mu_{i+1}}} + \frac{\bar{\pi}}{1 - \beta} \|q_{\mu_i}^{\mu_{i+1}} - q_{\mu_i}^{\mu_i}\|_1 < \tilde{\epsilon}.$$

Note that in line 5 of Algorithm 3 we do not intend to pre-compute μ_{i+1} ; this is intractable. Rather, we simply update the inputs to an oracle M that will be called upon to compute μ_{i+1} at sampled states in the subsequent iteration.

While Section 4 does suggest that if the approximation architecture we have chosen, Φ , is indeed a good approximation to the value function at a candidate equilibrium strategy, then Algorithm 3 will yield a good approximation to an MPE, there are no a-priori guarantees of this. Selecting a good approximation architecture is a problem specific task. In Section 6 we introduce an architecture that appears to be effective

approximating architecture, 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 players that will transition to state j from states $j - 1, j$, and $j + 1$.

Algorithm 3 Algorithm for Approximating MPE

```

1:  $\mu_0 := \mu$ 
2:  $i := 0$ 
3: repeat
4:    $r_i := \tilde{G}(\mu_i, \epsilon, L, \nu)$ 
     {Use Algorithm 2 with inputs  $\mu_i, \epsilon, L$  and  $\nu$ }
5:    $\mu_{i+1} := M(r_i, r_{i-1}, \dots, r_0, \mu_0)$ 
     {Update inputs to the oracle  $M$  so as to be able to compute  $\mu_{i+1}$  at sampled states}
6:    $\Delta = \|V_{\mu_i}^{\mu_{i+1}} - V_{\mu_i}^{\mu_i}\|_{1, q_{\mu_i}^{\mu_{i+1}}} + \frac{\tilde{\pi}}{1-\beta} \|q_{\mu_i}^{\mu_{i+1}} - q_{\mu_i}^{\mu_i}\|_1$ 
7:    $i := i + 1$ 
8: until  $\Delta < \tilde{\epsilon}$ 

```

for a class of EP-type models. Indeed, that section is devoted to a thorough computational study of the efficacy of Algorithm 3 in approximating MPE for EP-type models with a large number of players that preclude exact methods. Before, we specify the oracle $M(\cdot)$ and show that it can efficiently compute μ_{i+1} at any specific state.

5.3 Computing Strategies given a sequence of weight vectors: the oracle M

From Remark 5.2, it may appear that the computational complexity of a call to oracle M grows exponentially with each iteration of our proposed Algorithm 3 above. Fortunately, this is not the case as we now illustrate.

Fix $(x, s) \in \mathcal{X} \times \mathcal{S}$, and define $\mathfrak{N}(x, s)$ as the set of possible states faced by competing players, i.e.,

$$\mathfrak{N}(x, s) = \{(y, z) \in \mathcal{X} \times \mathcal{S} : s + e_x = z + e_y\},$$

where $e_i \in \mathbb{R}^{|\mathcal{X}|}$ is the i -th unit vector. Note that $|\mathfrak{N}(x, s)| = \Theta(N)$ for all x, s . We make an important observation at this juncture: for all $(y, z) \in \mathfrak{N}(x, s)$, $\mathfrak{N}(y, z) = \mathfrak{N}(x, s)$.

Now, let us say that at the i th iteration of Algorithm 3, we are required to compute $\mu_i(x, s)$ for some state (x, s) . We have that:

$$\mu_i(x, s) \in \operatorname{argmax}_{a \in \mathcal{A}^\epsilon} \pi(x, s, a) + \beta E_{\mu_{i-1}}[(\Phi r_{i-1})(x_1, s_1) | x_0 = x, s_0 = s, a_0 = a].$$

In addition to knowing r_{i-1} which is easy to store, this requires we compute $\mu_{i-1}(y, z)$ for all states $(y, z) \in \mathfrak{N}(x, s)$. But, unless $i - 1 = 0$, computing $\mu_{i-1}(y, z)$ will in turn require knowledge of r_{i-2} and $\mu_{i-2}(y', z')$ for all $(y', z') \in \mathfrak{N}(y, z)$. It thus appears that this level of the recursive procedure requires $\Theta\left(\sum_{(y, z) \in \mathfrak{N}(x, s)} |\mathfrak{N}(x, s)| |\mathfrak{N}(y, z)|\right) = \Theta(N^2)$ computations (up from $\Theta(N)$ at the first level of the recursion). However, as we previously observed, $\mathfrak{N}(y, z) = \mathfrak{N}(x, s)$, for all $(y, z) \in \mathfrak{N}(x, s)$. Therefore, the set

of states we must compute actions for at each level of the recursion does not grow; that is, the second level of the recursion will still require $\Theta(N)$ computations. This prevents the computation from blowing up; the computational complexity required at each stage of this recursive computation continues to remain $\Theta(N)$. More formally, Algorithm 4 computes $\mu_i(x, s)$ given a state (x, s) , basis function weights r_0, r_1, \dots, r_{i-1} and a strategy with an implicit compact representation μ .

Algorithm 4 $M(r_{i-1}, r_{i-2}, \dots, r_0, \mu_0)$ (Computation of $\mu_i(x, s)$ for $(x, s) \in \mathcal{X} \times \mathcal{S}$)

```

1:  $\mu_0 := \mu, j := 1$ 
2: repeat
3:   for all  $(y, z) \in \mathfrak{N}(x, s)$  do
4:      $\mu_j(y, z) := \operatorname{argmax}_{a \in \mathcal{A}^e} \pi(y, z, a) + \beta E_{\mu_{j-1}}[(\Phi r_{j-1})(x_1, s_1) | x_0 = y, s_0 = z, a_0 = a]$ .
5:      $j := j + 1$ 
6:   end for
7: until  $j = i$ 

```

Note that it is convenient to implement the algorithm by computing, in order, the actions under μ_1, μ_2 up to μ_i . It is worth confirming that the above is indeed an efficient algorithm. In particular, we note that since $|\mathfrak{N}(x, s)| \leq N$, Algorithm 4 makes $\Theta(i|N|)$ calls to line 4 in computing $\mu_i(x, s)$. Hence, the computational effort increases only linearly in the number of iterations. In the Appendix we present an alternative to the oracle M for which the computational effort does not increase with the number of iterations. This requires the solution of an alternative to the ALP that demands a more complex approximation architecture.

6 Computational Experiments

In this section we conduct computational experiments to evaluate the performance of our algorithm in situations where either we can compute an MPE, or a good approximation is available. Specifically, we compare the strategy derived from our algorithm against MPE for instances with relatively small state spaces, and against oblivious equilibrium for instances with large numbers of firms and parameter regimes where OE is known to provide a good approximation. We begin by specifying the EP-type model to be analyzed. The model is similar to Pakes and McGuire (1994) and Weintraub, Benkard, and Van Roy (2008a). However, it differs in that we consider a fixed number of firms (no entry or exit is permitted), and that we do not consider an aggregate shock that is common to all firms. Extending the approach to those models is straightforward. Then, in Section 6.2 we propose an approximation architecture. Finally, in Section 6.3 we report our numerical results.

6.1 The Computational Model

SINGLE-PERIOD PROFIT FUNCTION. We begin with deriving the single period profit function, π , from an underlying set of primitives describing the relevant industry. 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:

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

where Y is the consumer's income, p_{it} is the price of the good produced by firm i , and Z is a scaling factor. η_{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 zero utility. 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 classical logit model.

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

$$\sigma(x_{it}, s_{-i,t}, p_t) = \frac{M(x_{it}, p_{it})}{1 + \sum_{j \in S_t} M(x_{jt}, p_{jt})}, \quad \forall i \in S_t.$$

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,

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

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

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

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. Combining the investment and depreciation and appreciation processes,

it follows that the transition probabilities for a firm in state x that invests ι are given by:

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

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 marginal cost at $c = 0.5$, 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 an 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.⁸

6.2 A Separable Approximation Architecture

A key question of our approach is the selection of an approximation architecture. 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 . In particular, we would like to use an approximation of the form:

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

where $f_x^j : \{0, \dots, N - 1\} \rightarrow \mathbb{R}$ is an arbitrary univariate function specified for each pair $(x, j) \in \mathcal{X} \times \mathcal{X}$. 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 players at a particular state $j \in \mathcal{X}$. More sophisticated architectures, that, for example, explicitly capture interaction effects between numbers of players at different states, could be considered. However, this simple architecture appears to produce effective approximations in a class of EP models as is borne out in the computational experiments we present below.

We encode this approximation architecture using the following set of basis functions. Define the indica-

⁸To 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 1.25%.

tor function:

$$\phi_{i,j,k}(x, s) = \begin{cases} 1 & \text{if } x = i \text{ and } s(j) = k \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } (x, s) \in \mathcal{X} \times \mathcal{S}, k \in \{0, 1, \dots, N-1\}.$$

The basis matrix Φ is thus a matrix in $\mathbb{R}^{|\mathcal{X} \times \mathcal{S}| \times |\mathcal{X}| \cdot |\mathcal{X}| \cdot N}$ whose columns are functions of the type $\phi_{i,j,k}(\cdot, \cdot)$. Given a vector of weights $r \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{X}| \cdot N}$, 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 $|\mathcal{X}| \cdot |\mathcal{X}| \cdot N$ will typically be substantially smaller than $|\mathcal{X} \times \mathcal{S}|$, the use of this approximation architecture makes (4.9) a tractable program.

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 $s \in \{0, \dots, N-1\}$.⁹ Our selection of basis functions may be viewed as a generalization of approximation architectures that have been previously used in large scale stochastic control problems that arise in macroeconomics (Krusell and Smith 1998). Viewing $s/(N-1)$ as the p.m.f of some random variable S with support on $\mathcal{X} = \{0, \dots, \bar{x}\}$, these approximations consider the first few moments of the p.m.f. of S . For example, in this spirit, one could consider the following approximation architecture:

$$V_\mu^{\bar{\mu}}(x, s) \sim a_1(x)E[S] + a_2(x)E[S^2],$$

where one seeks functions $a_1(x)$ and $a_2(x)$ so as to produce a good approximation to $V_\mu^{\bar{\mu}}(x, s)$. It is easily seen that any approximation of this form is captured within our architecture. In fact, all approximations of the form $\sum_{i=1}^m a_i(x)E[S^i]$ must lie in the span of Φ . In our computational experiments we observe that

⁹In 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 program (4.9) as follows: For a set $\mathcal{N} \subseteq \{0, \dots, N\}$ define $l(n) = \max\{i \in \mathcal{N} : i \leq n\}$ and $u(n) = \min\{i \in \mathcal{N} : i \geq n\}$. We impose the following set of additional constraints:

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

That is, for each $i \in \mathcal{X}, j \in \mathcal{X}$, and $k \notin \mathcal{N}$, the variables $r_{i,j,k}$ are determined by linear interpolation.

moving beyond simple linear combinations of the, say, first two moments of S is valuable: 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. 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 industry state under the strategy in question. The indicators we examine are those that are typically of most interest to economists; we consider average investment, average producer surplus, average consumer surplus, average share of the largest firm (C1), and average share of the two largest firms (C2).

First, for instances with relatively small state spaces, we compute these indicators for the ALP-based strategy and compare them to the ones computed, under the same industry primitives, for the exact MPE strategy. Second, for instances with large numbers of firms and with choices of parameters for which OE provides accurate approximations to MPE, we compare the ALP indicators to the ones computed under OE strategies.¹⁰

For our computational experiments we use the relaxed stopping criterion $\Delta = \|V_{\mu_i}^{\mu_{i+1}}\|_{q_{\mu_{i+1}}} - \|V_{\mu_i}^{\mu_i}\|_{q_{\mu_i}}$. One can check that this quantity is always smaller than the criterion suggested on Section 3. Our computational experiments suggest that this criterion is effective to identify convergence to ϵ -weighted MPE. Also, 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. The parameter γ was set after some experimentation equal to $2/3$ to speed up convergence.

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 an MPE strategy. We compute MPE with Algorithm 1, solving program (4.3) to compute the operator F .¹¹

¹⁰Since 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 10^5 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%.

¹¹We also discretize actions to compute MPE.

We consider two different settings. In the first, there is a relatively rich investment process and the invariant distribution of the firm’s quality level tends to have a symmetric distribution around $\bar{x}/2$. We set $\bar{x} = 10$, $Z = 1$, $\theta_1 = 0.5$, $m = 75$ and $d = 3$. Table 1 reports the long-run statistics for $N = 2$ to $N = 6$; in these instances 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 took less than 5 minutes on each run. Exact computation of MPE took from a couple of seconds, for $N = 2$, to several hours, for $N = 6$.¹²

Note that ALP-based indicators are always within 2% from MPE indicators, and often within 0.5%. In particular, we observe that the magnitude of the differences does not change much across the different scenarios.

Number of Firms	Long-Run Statistics					
		Total Inv.	Prod. Surp.	Cons. Surp.	C1	C2
$N = 2$	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

Table 1: Comparison of MPE and ALP-based indicators for an industry with high investment levels. Long-run statistics computed simulating industry evolution over 10^5 periods.

Second, we consider an industry where incentives to invest are weaker and the invariant distribution of the firm’s quality level tends to be skewed, reflecting low levels of investment. We set $\bar{x} = 20$, $Z = 0.5$, $\theta_1 = 1.0$, $m = 10$ and $d = 2.5$. Table 2 reports the long-run statistics for $N = 2$ to $N = 4$; in these instances 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 took less than 5 minutes in each run

¹²All 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.

and exact computation of MPE took from a couple of seconds, for $N = 2$, to a couple of hours, for $N = 4$.

Number of Firms	Long-Run Statistics					
		Total Inv.	Prod. Surp.	Cons. Surp.	C1	C2
$N = 2$	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

Table 2: **Comparison of MPE and ALP-based indicators for an industry with low investment levels.** Long-run statistics computed simulating industry evolution over 10^5 periods.

We see that ALP-based indicators are always within 1.5% from MPE indicators and often within 0.5%. The results show that our ALP-based algorithm produces a good approximation to MPE, in instances with relatively small state spaces for which MPE can be computed. Moreover, our ALP-based algorithm requires substantially less computational effort.

6.3.2 Comparison with Oblivious Equilibrium

For large state spaces an 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 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 (2008b) establishes conditions under which OE well-approximates MPE asymptotically as the number of firms grows. Weintraub, Benkard, and Van Roy (2008a) 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 (2008a) 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 thousand 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.

Again, we consider two different settings. First, we consider an industry with a ‘low level of vertical differentiation’ (θ_1 is small) and where investment is not expensive. We set $\bar{x} = 20$, $Z = 1$, $\theta_1 = 0.3$ and $d = 0.6$. In this setting, computing an MPE using Algorithm 1 is infeasible for $N > 4$. We consider the pairs $(m, N) = \{(200, 20), (300, 30)\}$. Under these configurations, the bounds derived on Weintraub, Benkard, and Van Roy (2008a) suggest that OE indicators provide a good approximation to MPE. We label these configurations as (m, N, L) , for ‘low level of vertical differentiation’.

Next, we consider an industry with a higher level of vertical differentiation, where θ_1 is larger, and investment is more expensive. We set $\bar{x} = 20$, $Z = 2$, $\theta_1 = 0.5$ and $d = 0.8$. For this setting we need more firms to achieve acceptable OE error bounds. Therefore, we consider the pair $(m, N) = (500, 50)$. This configuration is labeled as $(500, 50, H)$ for ‘high level of vertical differentiation’.

Table 3 reports the long-run statistics for these configurations. There, OE and ALP-based long-run statistics values are reported, together with the percentage differences between them. Our algorithm took from 4 to 8 hours in each run and OE computation took less than 5 minutes for each instance.

N. of Firms / Marker size / Level of vert diff	Long-Run Statistics					
		Total Inv.	Prod. Surp.	Cons. Surp.	C1	C2
(20, 200, L)	OE	9.2320	63.4985	556.0570	0.0578	0.1117
	ALP-Based	9.4844	63.1420	538.5704	0.0561	0.1108
	% Diff.	2.73	0.56	3.14	2.94	0.73
(30, 300, L)	OE	13.9668	96.8077	953.1851	0.0397	0.0775
	ALP-Based	14.4277	96.5319	929.5035	0.0383	0.0758
	% Diff.	3.30	0.28	2.48	3.58	2.25
(50, 500, H)	OE	21.4580	164.1466	1924.0496	0.0274	0.0539
	ALP-Based	22.2574	163.6019	1847.1416	0.0282	0.0547
	% Diff.	3.73	0.33	4.00	3.15	1.46

Table 3: **Comparison of OE and ALP-based indicators.** OE statistics simulated with a relative precision of 1.0% and a confidence level of 99%.

We observe that ALP-based indicators are always within 4% from OE indicators. 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 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. However, we believe that the numerical experiments described above suggest that our method should provide effective approximations to MPE there as well, 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 stochastic games. 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-type 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 solving problems that, given currently available methods, have to this point been infeasible.

Commonly, EP-type models allow for exit decisions by incumbent firms and for entry decisions by potential entrants. They also include aggregate shocks that are common to all firms. These elements make the dynamics more realistic. The EP model we study in this paper does not include those features. However, extending our method to allow for entry and exit decisions, and aggregate shocks is straightforward. We plan to experiment with these extensions in future work.

Finally, an important contributor to the success of our approach is the selection of a good approximation architecture. In this paper, we showed that a simple separable architecture is effective for the class of EP-type 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 for other classes of dynamic stochastic games. In this way, we hope that our method will also find applicability in contexts beyond EP models.

References

- Basar, T. and J. Olsder (1999). *Dynamic Noncooperative Game Theory* (Second ed.). SIAM.
- Benkard, C. L. (2004). A dynamic analysis of the market for wide-bodied commercial aircraft. *Review of Economic Studies* 71(3), 581 – 611.

- 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.
- Borkovsky, R., U. Doraszelski, and Y. Kryukov (2008). A user's guide to solving dynamic stochastic games using the homotopy method. Working Paper, Harvard University.
- Breton, M. (1991). Algorithms for stochastic games. In *T.E.S. Raghavan, T.S. Ferguson, T. Parthasarathy, O.J. Vrieze (Eds.), Stochastic Games and Related Topics: In Honor of Professor L.S. Shapley*, pp. 45 – 57.
- Caplin, A. and B. Nalebuff (1991). Aggregation and imperfect competition - on the existence of equilibrium. *Econometrica* 59(1), 25 – 59.
- Collard-Wexler, A. (2006). Productivity dispersion and plant selection in the ready-mix concrete industry. Working Paper, NYU.
- de Farias, D. P. and B. V. Roy (2003). The linear programming approach to approximate dynamic programming. *Operations Research* 51(6), 850 – 865.
- de Farias, D. P. and B. V. 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.
- Escobar, J. (2006). Time-homogenous markov equilibrium in dynamic stochastic games. Working Paper, Stanford University.

- Filar, J., T. Schultz, F. Thuijsman, and O. Vrieze (1991). Nonlinear programming and stationary equilibria in stochastic games. *Mathematical Programming* 50, 227 – 237.
- Filar, J. and K. Vrieze (1997). *Competitive Markov Decision Processes*. Springer.
- Fudenberg, D. and D. K. Levine (1998). *The Theory of Learning in Games*. MIT Press.
- Fudenberg, D. and J. Tirole (1991). *Game Theory*. MIT Press.
- Govindan, S. and R. Wilson (2008). Global newton method for stochastic game. Forthcoming, *Journal of Economic Theory*.
- Gowrisankaran, G. and R. Town (1997). Dynamic equilibrium in the hospital industry. *Journal of Economics and Management Strategy* 6(1), 45 – 74.
- Herings, P. and R. Peeters (2004). Stationary equilibria in stochastic games: structure, selection, and computation. *Journal of Economic Theory* 118, 32 – 60.
- Jenkins, M., P. Liu, R. L. Matzkin, and D. L. McFadden (2004). The browser war - econometric analysis of Markov perfect equilibrium in markets with network effects. Working Paper.
- 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.
- Perakis, G., S. Kachani, and C. Simon (2008). Closed loop dynamic pricing under competition. Working Paper, MIT.
- Ryan, S. (2005). The costs of environmental regulation in a concentrated industry. MIT, Mimeo.
- Shapley, L. (1953). Stochastic games. *Proceedings of the National Academy of Sciences* 39, 1095 – 1100.
- 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 (2008a). Computational methods for oblivious equilibrium. Working Paper, Stanford University.

Weintraub, G. Y., C. L. Benkard, and B. Van Roy (2008b). Markov perfect industry dynamics with many firms. Forthcoming, *Econometrica*.

A Proofs

Theorem 3.1. *Let $\{\mu_n \in \mathcal{M} | n \in \mathbb{N}\}$ be a sequence of ϵ_n -weighted MPE. Suppose that $\lim_{n \rightarrow \infty} \epsilon_n = 0$. Then, $\lim_{n \rightarrow \infty} d(\Gamma, \mu_n) = 0$.*

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

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

$$(A.2) \quad \mu'_n \rightarrow \mu^*.$$

$$(A.3) \quad \inf_n d(\Gamma, \mu'_n) > \epsilon,$$

where $\mathcal{BR}(\mu)$ denotes the best response strategy when competitors play strategy μ . Now (A.2) and Assumption 2.1.3 imply that $q_{\mu'_n}^{\mu'_n} \rightarrow q_{\mu^*}^{\mu^*}$. In addition,

$$0 \leq \|V_{\mu'_n}^{\mathcal{BR}(\mu'_n)} - V_{\mu'_n}^{\mu'_n}\|_{1, q_{\mu^*}^{\mu^*}} \leq \frac{\bar{\pi}}{1 - \beta} \|q_{\mu^*}^{\mu^*} - q_{\mu'_n}^{\mu'_n}\|_1 + \|V_{\mu'_n}^{\mathcal{BR}(\mu'_n)} - V_{\mu'_n}^{\mu'_n}\|_{1, q_{\mu'_n}^{\mu'_n}}.$$

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

$$\|V_{\mu'_n}^{\mathcal{BR}(\mu'_n)} - V_{\mu'_n}^{\mu'_n}\|_{1, q_{\mu^*}^{\mu^*}} \rightarrow 0.$$

Now since by Assumption 2.1, we must have $q_{\mu^*}^{\mu^*} > 0$ component-wise, this implies that component-wise

$$(A.4) \quad V_{\mu_n}^{\mathcal{BR}(\mu_n')} - V_{\mu_n}^{\mu_n'} \rightarrow 0.$$

Now, by Assumption 2.1 and Berge's maximum theorem, $\mathcal{BR}(\cdot)$ is continuous on \mathcal{M} and V_{μ}^{μ} is continuous in μ . Thus, we have from (A.4) and (A.2) that $V_{\mu^*}^{\mathcal{BR}(\mu^*)} - V_{\mu^*}^{\mu^*} = 0$ so that $\mu^* \in \Gamma$. But by (A.3) and the triangle inequality $d(\Gamma, \mu^*) > \epsilon$, a contradiction. The result follows. \square

Theorem 4.3. *Let $\tilde{\epsilon} < 1$ satisfy $1 - \tilde{\epsilon} \leq \frac{\mathcal{P}(f(x, \underline{a} + \lfloor (a - \underline{a})/\epsilon \rfloor \epsilon, \zeta) = x')}{\mathcal{P}(f(x, a, \zeta) = x')}$, $\forall x, x', a$. Moreover, let $\pi(x, s, \cdot)$ have Lipschitz constant K for all x, s . Finally, let r_{μ}^{ϵ} be an optimal solution to (4.8). Then:*

$$\|\Phi r_{\mu}^{\epsilon} - V_{\mu}^*\|_{1,c} \leq \frac{2}{1 - \beta} \inf_r \|\Phi r - V_{\mu}^*\|_{\infty} + \frac{3 - \beta}{1 - \beta} \left(\frac{2\tilde{\epsilon}\beta\bar{\pi}}{(1 - \beta)^2} + \frac{K\epsilon}{1 - \beta} \right).$$

Proof. Let us denote by $V_{\mu}^{*,\epsilon}$ the value function corresponding to a best response strategy to μ when actions in a given time are restricted to the set \mathcal{A}^{ϵ} . We begin showing that

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

Let $P_{\mu}^* \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{S}| \times |\mathcal{X}| \times |\mathcal{S}|}$ be a state transition matrix corresponding to using the best response strategy μ^* in response to μ . Define μ^{ϵ} according to $\mu^{\epsilon}(x, s) = \underline{a} + \lfloor (\mu^*(x, s) - \underline{a})/\epsilon \rfloor \epsilon$ and let P_{μ}^{ϵ} be the corresponding state transition matrix. Moreover, let $g, g^{\epsilon} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{S}|}$ be respectively defined according to $g(x, s) = \pi(x, s, \mu^*(x, s))$, $g^{\epsilon}(x, s) = \pi(x, s, \mu^{\epsilon}(x, s))$. Now since $1 - \tilde{\epsilon} \leq \frac{\mathcal{P}(f(x, \underline{a} + \lfloor (a - \underline{a})/\epsilon \rfloor \epsilon, \zeta) = x')}{\mathcal{P}(f(x, a, \zeta) = x')}$, $\forall x, x', a$ by assumption, we must have that

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

for some 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 $\tilde{V}_{\mu}^* = \sum_{t=0}^{\infty} \beta^t P_{\mu}^{*t} g^{\epsilon}$, and noting that the maximal absolute difference in the performance of two arbitrary strategies starting from a given state is bounded from above by $\frac{2\bar{\pi}}{1 - \beta}$, this lets us conclude that

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

Now by definition and Lipschitz continuity,

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

Thus, by the triangle inequality,

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

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

$$(A.5) \quad 0 \leq V_\mu^* - V_\mu^{*,\epsilon} \leq \frac{2\tilde{\epsilon}\beta\bar{\pi}}{(1-\beta)^2} + \frac{K\epsilon}{1-\beta}.$$

Now, we know from Theorem 4.1 that

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

With the triangle inequality and (A.5) this yields

$$\begin{aligned} \|\Phi r_\mu^\epsilon - V_\mu^*\|_{1,c} &\leq \|\Phi r_\mu^\epsilon - V_\mu^{*,\epsilon}\|_{1,c} + \|V_\mu^{*,\epsilon} - V_\mu^*\|_{1,c} \\ &\leq \frac{2}{1-\beta} \inf_r \|\Phi r - V_\mu^{*,\epsilon}\|_\infty + \left(\frac{2\tilde{\epsilon}\beta\bar{\pi}}{(1-\beta)^2} + \frac{K\epsilon}{1-\beta} \right) \\ &\leq \frac{2}{1-\beta} \inf_r \|\Phi r - V_\mu^*\|_\infty + \frac{3-\beta}{1-\beta} \left(\frac{2\tilde{\epsilon}\beta\bar{\pi}}{(1-\beta)^2} + \frac{K\epsilon}{1-\beta} \right). \end{aligned}$$

□

B An Approximate LP for Q-functions

We present here an alternative to the approximate linear program that will alleviate the computational complexity introduced by the oracle M ; in particular, we present an approach to computing an approximate best response whose complexity does not grow with each iteration of our best response computation.

For a fixed $\mu \in \mathcal{M}$ and an arbitrary $V \in \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$ define for every $a \in \mathcal{A}$,¹³ the operator $\bar{T}_\mu^a : \mathbb{R}^{|\mathcal{X} \times \mathcal{S}|} \rightarrow$

¹³To avoid discussing action space discretization in this brief presentation, we will assume \mathcal{A} is finite.

$\mathbb{R}^{|\mathcal{X} \times \mathcal{S}|}$ according to

$$(\bar{T}_\mu^a V)(x, s) = \pi(x, s, a) + \beta E_\mu[V(x_1, s_1) | x_0 = x, s_0 = s, a_0 = a], \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S},$$

With an abuse of notation, define for a given V , the ‘Q’-function, $Q_\mu : \mathcal{X} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ according to $Q_\mu(x, s, a) = (\bar{T}_\mu^a V)(x, s)$. Let $Q_\mu^*(x, s, a) = (\bar{T}_\mu^a V_\mu^*)(x, s)$ and observe that the best response strategy to μ is given by $\mu^*(x, s) = \operatorname{argmax}_{a \in \mathcal{A}} Q_\mu^*(x, s, a)$. We next present a program that simultaneously computes V_μ^* and Q_μ^* :

$$\begin{aligned} \text{(B.1)} \quad & \min \quad c'_1 V + c'_2 Q \\ & \text{s.t.} \quad (T_\mu V)(x, s) \leq V(x, s) \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S} \\ & \quad (\bar{T}_\mu^a V)(x, s) \leq Q(x, s, a) \quad \forall (x, s, a) \in \mathcal{X} \times \mathcal{S} \times \mathcal{A}. \end{aligned}$$

One may show that provided c_1 and c_2 are component-wise positive, (V_μ^*, Q_μ^*) is the unique optimal solution to the above program. Moreover, it is easy to rewrite the above program as a linear program. As in the case of the approximate LP we next consider approximating both V_μ^* as well as Q_μ^* . In particular, letting Φ_1, Φ_2 be basis matrices of appropriate dimension satisfying $\inf_{r_1} \|V_\mu^* - \Phi_1 r_1\|_\infty < \epsilon, \inf_{r_2} \|Q_\mu^* - \Phi_2 r_2\|_\infty < \epsilon$, consider the program

$$\begin{aligned} \text{(B.2)} \quad & \min \quad c'_1 \Phi_1 r_1 + c'_2 \Phi_2 r_2 \\ & \text{s.t.} \quad (T_\mu \Phi_1 r_1)(x, s) \leq (\Phi_1 r_1)(x, s) \quad \forall (x, s) \in \mathcal{X} \times \mathcal{S} \\ & \quad (\bar{T}_\mu^a \Phi_1 r_1)(x, s) \leq (\Phi_2 r_2)(x, s, a) \quad \forall (x, s, a) \in \mathcal{X} \times \mathcal{S} \times \mathcal{A}. \end{aligned}$$

One may then establish that an optimal solution (r_1^*, r_2^*) to the above program satisfies $\|Q_\mu^* - \Phi_2 r_2^*\|_\infty = O(\epsilon)$ and moreover that the greedy policy given by $\tilde{\mu}(x, s) = \operatorname{argmax}_a (\Phi_2 r_2^*)(x, s, a)$ has value that is within $O(\epsilon)$ of the optimal value. Of course, to render (B.2) tractable, we will have to resort to a similar constraint sampling procedure as we did in the case of the approximate LP.

One may consider employing a program such as (B.2) in place of (4.4) in computing an approximate best response at each iteration of our best response algorithm. The advantage to doing so would be that the oracle to compute competitor strategies at each iteration is substantially simplified: one simply requires the weights r_2^* computed at the prior iteration; the corresponding strategy is simply given by $\tilde{\mu}(x, s) = \operatorname{argmax}_a (\Phi_2 r_2^*)(x, s, a)$. The advantage to such an approach is that the computational complexity per iteration of our iterative best response scheme does not grow with each iteration. The drawbacks are that the program above has a larger number of variables, and one now requires access to a suitable approximation architecture

to approximate Q^* .