Finite State Dynamic Games with Asymmetric Information: A Framework for Applied Work. <<Preliminary Version>>

Chaim Fershtman and Ariel Pakes

April 2007.

Abstract

We present a simple algorithm for computing an intuitive notion of equilibrium for finite state dynamic games with asymmetric information. The algorithm does not require; storage and updating of posterior distributions, explicit integration over possible future states to determine continuation values, or storage and updating of information at all possible points in the state space. It is also easy to program. We conclude with an example that endogenizes the maintenance decisions for electricity generators in a dynamic game among electric utilities in which the costs states of the generators are private information.

This paper develops a relatively simple framework for the applied analysis of dynamic games with asymmetric information. We consider a class of dynamic games in which there are a finite number of active players in each period, each characterized by a vector of state variables. Some of these state variables are publicly observable while others are private information. State variables evolve over time with the outcome of the players' actions. In addition to affecting the evolution of the state variables these actions may provide signals on the variables that are private information.

We design our framework so that there are only a finite number of possible states of the game. In this context we provide an equilbrium concept whose conditions are, at least in principal, testable and do not require computation of posterior distributions. We then provide an algorithm which computes equilibria of the game. The algorithm consists of an iterative procedure and a test of the equilibrium conditions, neither of which are subject to a curse of dimensionality. We conclude with an example that endogenizes the maintenance decisions for electricity generators in a dynamic game among electric utilities in which the costs states of the generators are private information.

In the game we consider the number of active players may change over time due to entry and exit. Each active player is characterized by a vector of state variables (e.g. indexes of their cost function, qualities of the goods they market, Network size, capacity etc.) which can take values only on a finite space. The state variables of one firm are not necessarily observable to the other firms. Each firm's returns in a given period are determinned by the firms' state variables and their actions. The actions are allowed to contain a set of continuous controls (e.g. investment), and a set of discrete controls (e.g. sending a signal, entry, exit, etc.). The discrete control may or may not be observable to the firms' competitors and are taken from a finite set. The continuous controls are not observable to the firms' competitors and affects the game through their impact on the probability distribution of discrete state variables. In addition to these conditions we require a restriction on the informational structure of the game to guarantee that the state space be finite (there are alternative possiblities here, see below).

We define an Applied Markov Equilibrium for our game as a triple which satisfies three conditions. The triple consists of; (i) a vector of strategies defined for every possible information set of each agent, (ii) a subset of the set of possible states, and (iii) a vector of values for every state that provides the expected discounted value of net cash flow conditional on the possible outcomes of the agent's actions at that state. The first condition is that

the subset of states is a recurrent class of the Markov process generated by the equilibrium strategies. The second condition is that the strategies are optimal given the evaluations of outcomes for all points in this recurrent class, and the last condition is that these evaluations are indeed the expected discounted value of future net cash flows on the recurrent class if all agents play the equilibrium strategies.

When all the state variables are observed to all the agents, our equilibrium notion is similar to but weaker than the familiar notion of Markov Perfect equilibrium as used in Maskin and Tirole (1986a, 1986b). This because we only require that the evaluations be consistent with the outcomes of observed play on the recurrent class. In this respect our equilibrium concept is related to the notion of self-confirming equilibrium, as defined by Fudenberg and Levine (2001).

There are two reasons our notion of equilibrium might be suitable for applied work. First since the only points that will be observed infinitely often from equilibrium points are points in the recurrent class.

Methodologically we show that in games of this form there are simple sufficient statistics such that if agents maximizes with respect to them, then their actions will be Applied Markov Equilibrium policies. These sufficient statistics are the expected discounted value of future net cash flows given the possible outcomes of their choice of controls. These discounted values are computed conditional on the firms' information sets, and these, in turn, define the possible states of the game.

If one knew the empirical distribution of outcomes from each state, one could compute the needed sufficient statistics directly from that distribution and the primitives of the problem. As a result any agent playing the game who had a history of outcomes at its disposal could calculate these statistics by computing averages. Thus players can determine their optimal behavior from a relatively simple set of calculations; in particular the players never have to bother with computing posterior beliefs for the states of their competitors (at least on the set of states that are visited repeatedly or, more formally, visited "infinitely often").

What we show is that when this empirical distribution is unknown, there is a reinforcement learning (or stochastic approximation) algorithm which enables one to compute the equilibrium by a procedure which only requires updating averages (in particular it never requires one to compute posterior

distrubutions or to integrate out over possible future values)[1]. Thus either the players themselves, or a computer algorithm, could use random outcomes and a set of simple calculations to compute the sufficient statistics which determine equilibrium play (again, at least play on a recurrent class of points). That is the reinforcement learning algorithm converts the seemingly intractable problem of computing the Applied Markov Equilibrium into a relatively simple problem of updating averages (so simple that we have computed our examples on a five year old laptop computer).

We only expect the algorithm to provide the correct sufficient statistics as the number of iterations increases without bound. However the paper provides a stopping rule which checks whether the algorithm has produced sufficient statistics within any given accuracy of their equilibrium values, and hence can be used to determine when an adequate approximation has been found[2].

The next section describes the details of the game in a general setting and provides a definition of, and sufficient conditions for, a Markov Perfect equilibrium to this game. Section 2 provides an algorithm which computes the equilibrium to this game. Section 3 introduces our example, section 4 provides the details of how to compute its equilibrium, and section 5 provides numerical results from computing it and related models.

[1]The stochastic approximation literature dates back to the classic paper of Robbins and Monroe,1956, and has been used extensively for calculating solutions to single agent dynamic programming problems (see Bertsekas and Tsikilis,1996 and the literature they cite). Pakes and McGuire,2001, show that it has significanat computational advanatages when applied to full information dynamic games, but as we will show the advantages in using it to compute the solution to assymetric information dynamic games are much larger.

[2]An alternative approach would have been to view the AI algorithm itself as the way players learn the statistics needed to choose their policies, and justify the output of the algorithm in that way.

1 A Finite State Dynamic Game with Asymmetric Information.

We extend the framework in Ericson and Pakes (1995) to allow for asymmetric information.[3] In each period there are n_t potentially active firms, and we assume that with probability one $n_t \leq \overline{n} < \infty$ (for every t). Each firm has payoff relevant characteristics. Typically these will be characteristics of the products marketed by the firm or of their cost functions. The profits of each firm in every period are determined by; the payoff relevant random variables of all of the firms, a subset of the actions (or controls) of all the firms, and a set of common determinants of demand and costs, say $d \in D$.

We only consider games with *finite state spaces* and to insure that this is the case we make the following assumptions.[4] The payoff relevant characteristics, which will be denoted by $\omega \in \Omega$, take values on a finite set of points. There will be two types of actions (or controls); actions which take values on a finite space, denoted by $m \in M$, and actions which take values on a continuum, to be denoted by $x \in X$. It will be assumed that the continuous action of one firm is neither observed by the other firm nor a determinant of the profits of the other firm (if this were not true then firms would have to condition on signals which take values on a continuum). However the discrete actions of the firm are not restricted in either of these two ways. Both the continuous and discrete action can affect current profits and/or the probability distribution of payoff relevant random variables.

For notational simplicity we will assume that the there is only one state variable, one discrete control, and one continuous control for each firm; i.e. that $\Omega \subset Z_+$, $M \subset Z_+$, and $X \subset R$. In different models both the actions and the states will have different interpretations. Possibilities for actions include; the sending of signals, entry, exit, bidding in an auction, the launching of new products, and so on.

Letting i index firms, realized profits for firm i in period t are given by

$$\pi(\omega_{i,t}, \omega_{-i,t}, m_{i,t}, m_{-i,t}, x_{i,t}, d_t), \tag{1}$$

[3] Since we were motivated by our interest in dynamic oligopolies we will call our players "firms" and their payoffs as "profits".

[4] Alternatively we could have allowed states to take values in a countable set but then invoked conditions which insure that we only observe values on a finite subset (for an example see Ericson and Pakes, 1995).

where $\pi(\cdot) : \Omega^n \times M^n \times R \times D \rightarrow R$. Note that though firms know their own (ω, x, m), we have not yet specified what they know about their competitors (ω, m).

We assume that $\omega_{i,t}$ evolves over time with random firm specific outcomes, to be denoted by $\eta_{i,t}$, and a common shock that affects the ω's of all firms in a given period, say ν_t. Both η and ν take on values in a finite subset of $\mathcal{Z}_+$, say in $\Omega(\eta), \Omega(\nu)$ respectively. The transition rule is written as

$$\omega_{i,t+1} = F(\omega_{i,t}, \eta_{i,t+1}, \nu_{t+1}), \tag{2}$$

where $F : \Omega \times \Omega(\eta) \times \Omega(\nu) \rightarrow \Omega$. The distribution of η is determined by the family

$$\mathcal{P}_\eta = \{ P_\eta(.|\, m, x, \omega); \;\; m \in M, \; x \in X, \;\; \omega \in \Omega\}, \tag{3}$$

while the distribution of ν is given exogenously and equal to

$$\{p(\nu); \nu \in \Omega(\nu)\}.$$

Note that at least in this formulation of our problem we do not allow either the states or the actions of a firm's competitors to effect the evolution of the firm's own state variables.

The information set of each player at period t is, in principal, the whole observable history up to that period. We restrict ourselves to a class of games that can be defined on a finite state space. The states themselves include, in addition to "payoff relevant" variables in Maskin and Tiorle's (2001) sense of variables that affect current profits, variables that are not determinants of the current profits but do provide information on the distribution of future profits. We will refer to these variables as "informationally relevant" variables. More precisely a variable is neither payoff nor informationally relevant in a given period only if, when each player's rivals do not condition there strategies on that variable, the player itself can not gain by conditioning its strategy on that variable. The requirement that the state space be finite implies that we limit ourselves to games where there are only a finite number of values for the informationally relevant variables as well as for the payoff relevant random variables. We consider examples that satisfy this restriction below.

For simplicity we limit ourselves to the case where information is either known only to a single agent (it is "private"), or to all agents (it is "public").

Different models will allocate different states and actions to the publicly and privately observed components. The publicly observed vector will be denoted by $\xi_t \in \Omega(\xi)$, and the privately observed vector by $z_{i,t} \in \Omega(z)$, with both $\#\Omega(\xi)$ and $\#\Omega(z)$ finite.

If both decisions and the evolution of states conditional on those decisions depend only on $(\xi_t, z_{i,t})$, $(\xi_t, z_{i,t})$ evolves as a Markov process. More formally for all t

$$\xi_{t+1} = G_\xi(\xi_t, \nu_{t+1}, \epsilon_{t+1}), \tag{4}$$

and

$$z_{i,t+1} = G_z(\xi_t, z_{i,t}, \mu_{t+1}), \tag{5}$$

where the distribution of ϵ_{t+1} is given by the family

$$\mathcal{P}_\epsilon = \{\ P_\epsilon(\cdot|\ \xi, x, m, z, \eta,\ (\xi, x, m, \eta, z) \in \Omega(\xi) \times (X \times M \times \Omega(\eta) \times \Omega(z))^n\ \}, \tag{6}$$

and the distribution of $\mu_{i,t+1}$ is given by the family

$$\mathcal{P}_\mu = \{\ P_\mu(\cdot|\ \xi, x, m, z, \eta),\ (\xi, x, m, \eta, z) \in \Omega(\xi) \times (X \times M \times \Omega(\eta) \times \Omega(z))^n\ \}. \tag{7}$$

Here ϵ_{t+1} contains the public information that is revealed over the current period, while $\mu_{i,t+1}$ contains the private information revealed to firm i over the current period.

Since the agent's information at the time actions are taken consists of $J_{i,t} = (\xi_t, z_{i,t}) \in \mathcal{J}_i$, we assume strategies are measurable $J_{i,t}$, i.e.

$$x(J_{i,t}) :\quad \mathcal{J}_i \to X,\ \text{ and }\ m(J_{i,t}) :\quad \mathcal{J}_i \to M.$$

The timing of the game is as follows. At the beginning of each period there is a realization of $\{\mu, \nu, \epsilon, \}$. Firm's then update their information sets with the updating functions (5) and (4). They then simultaneously decide on $\{m_{i,t}, x_{i,t}\}_{i=1}^{n_t}$. Finally we assume that firms maximize their expected discounted value of profits and have a common discount rate β , where $0 < \beta < 1$.

This formulation enables us to account for a range of institutional structures. The original Ericson and Pakes (1995) full information assumptions is the special case where $\xi_t = (\omega_{i,t}, \omega_{-i,t})$ and $\epsilon_t = (\eta_{i,t}, \eta_{-i,t})$. When there is asymmetric information ϵ represents the new information revealed to the public in a given period, and μ represents the increment in private information. The distribution of ϵ can depend on the actions of agents, as when

signals are sent, and its contents can depend on the private information of all agents in the preceeding period, on $\omega \in \Omega^n$, and/or on the private information obtained in the current period, on $\eta \in \Omega(\eta)^n$. A simple example of a game with private information occurs when $\xi_t = (\omega_{i,t-1}, \omega_{-i,t-1})$ and $\mu_{i,t} = \eta_{i,t}$, so $z_{i,t} = \omega_{i,t}$ (each firm knows the current value of its own ω but only last period's value of the ω's of its competitors). These are the games considered in the recent econometric literature on dynamic games (see Pakes, Ostrovsky and Berry, forthcoming, and Bajari, Benkard and Levin, forthcoming)[5].

2 An Applied Markov Equilibrium.

Let s combine the information sets of all agents active in a particular period, that is $s = (J_1, \ldots, J_n)$ when each J_i has the same public component ξ. We will say that $J_i = (z_i, \xi)$ is a component of s if it contains the information set of one of the firms whose information is combined in s. Note also that we can write s more compactly as $s = (z_1, \ldots, z_n, \xi)$. So $\mathcal{S} = \{s : z \in \Omega(z)^n, \xi \in \Omega(\xi), \text{ for } 0 \leq n \leq \overline{n}\}$ lists the possible states of the world.

Any set of Markov strategies for all agents active at each $s \in \mathcal{S}$, together with an initial condition, defines a Markov process on $\mathcal{S}$. Since $\mathcal{S}$ is a finite set, each possible sample path of this Markov process will, in finite time, wander into a recurrent subset of the states in $\mathcal{S}$, say $\mathcal{R} \subset \mathcal{S}$, and once in $\mathcal{R}$ will stay within it forever. That is a point, s, will be visited infinitely often if and only if $s \in \mathcal{R}$. Moreover the empirical distributions of visits to transitions in $\mathcal{R}$ will converge to a Markov transition kernel, say $p^{e,T} \equiv \{p^e(s'|s) : (s', s) \in \mathcal{R}^2\}$, while the empirical distribution of visits on $\mathcal{R}$ will converge to an invariant measure, say $p^{e,I} \equiv \{p^e(s) : s \in \mathcal{R}\}$. We let $p^e = (p^{e,T}, p^{e,I})$.

Our notion of Applied Markov Perfect equilibrium is that of an equilibrium whose conditions could, at least in principle, be consistently tested. To obtain a consistent test of a condition at a point we must, at least potentially, observe that point infinitely often. So we limit ourselves to tests of conditions at points in $\mathcal{R}$. That is we define an Applied Markov Perfect equilibrium by the set of equilibrium conditions which are consistent with optimizing behavior and p^e only in the recurent class derived by the equilibrium

[5]Note that whenever ω_{-i} has an independent effect on the profits of firm i (independent of other information known to the agent), and profits are privately observed, then μ_i at least contains (η_i, π_i).

strategies.

As we shall see this weakens the traditional Nash conditions. In fact it generates equilibria which are closely related to the "self-confirming equilibria" introduced by Fudenberg and Levine (1993a), and share some of that equilibria's interpretive advantages[6]. On the other hand ours is probably the strongest notion of equilibrium that one might think could be empirically tested, as it assumes that the applied researcher doing the testing can access the union of the information sets available to the agents playing the game. We come back to these issues, and their relationship to empirical work, after we provide our definition of equilibrium.

Definition: Applied Markov Equilibrium. An applied Markov Equilibrium consists of

- A subset $\mathcal{R} \subset \mathcal{S}$;
- Strategies $(x^*(J_i), m^*(J_i))$ for every J_i which is a component of any $s \in \mathcal{S}$;
- Expected discounted value of current and future net cash flow conditional on realizations of η and a value for the discrete decision m, say $W(\eta, m|J_i)$, for each $(\eta, m) \in \Omega(\eta) \times M$ and every J_i which is a component of any $s \in \mathcal{S}$,

such that

C1: $\mathcal{R}$ is a recurrent class. The Markov process generated by any initial condition $s_0 \in \mathcal{R}$, and the transition kernel generated by $\{(x^*, m^*)\}$, has $\mathcal{R}$ as a recurrent class (so, with probability one, any subgame starting from an $s \in \mathcal{R}$ will generate sample paths that are within $\mathcal{R}$ forever).

C2: Optimality of strategies on $\mathcal{R}$. For every J_i which is a component of an $s \in \mathcal{R}$, strategies are optimal given $W(\cdot)$, that is $(x^*(J_i), m^*(J_i))$ solve

$$max_{m \in M} sup_{x \in X} \left[\sum_{\eta} W(\eta, m|J_i) p_\eta(\eta|x, m, \omega_i) \right],$$

[6] See also Dekel, Fudenberg and Levine (2004) for an anlysis of self confirming equilibrium in games with asymmetric information.

and

C3: Consistency of values on $\mathcal{R}$. Take every J_i which is a component of an $s \in \mathcal{R}$. Let $\eta(x^*(J_i), m^*(J_i), J_i) \equiv \{\eta : p(\eta|x^*(J_i), m^*(J_i), \omega_i) > 0\}$. For every $\eta \in \eta(x^*(J_i), m^*(J_i), J_i)$

$$W(\eta, m^*(J_i)|J_i) = \pi^E\Big(J_i, m^*(J_i), x^*(J_i)\Big) \; + \;$$
$$\beta \sum_{J_i'} \Big\{ \sum_{\tilde{\eta}} W(\tilde{\eta}, m^*(J_i')|J_i') p(\tilde{\eta}|x^*(J_i'), m^*(J_i'), \omega_i') \Big\} p^e(J_i'|J_i, \eta),$$

and

$$\pi^E(J_i; m^*(J_i)) \equiv \sum_{J_{-i}} \pi_i\Big(\omega_i, m^*(J_i), J_{-i}, d_t\Big) p^e(J_{-i}|J_i),$$

where

$$\left\{ p^e(J_i'|J_i, \eta) \equiv \frac{p^e(J_i', \eta|J_i)}{p^e(\eta|J_i)} \right\}_{J_i'}, \text{ and } \left\{ p^e(J_{-i}|J_i) \equiv \frac{p^e(J_{-i}, J_i)}{p^e(J_i)} \right\}_{J_{-i}} . \spadesuit \quad (8)$$

Condition C2 states that at every J_i which is a component of an $s \in \mathcal{R}$ agent i choses policies which are optimal with respect to the evaluations of outcomes determined by $\{W(\eta, m|J_i) : \eta \in \Omega(\eta), m \in M\}$. Condition C3 states that at least for (η, m) combinations that have positive probability on the equilibrium path, these evaluations are the values that would be generated by p^e and the primitives of the problem if the agent played equilibrium strategies.

Note that conditions C2 and C3 apply only to points in $\mathcal{R}$. In particular policies at points outside of $\mathcal{R}$ need not be optimal while the evaluations $\{W(\eta, m|J_i)\}$ need not be correct for J_i not a component of an $s \in \mathcal{R}$. Nor do we require consistency of the evaluations for the $W(\cdot)$'s associated with points in $\mathcal{R}$ but outcomes which have zero probability given equilibrium play[7]. The fact that our conditions do not apply to points outside of $\mathcal{R}$ implies that the conditional probabilities in equation (8) are well defined.

[7] To see this last point note that the $W(\eta, m|J_i)$ for $\eta \notin \eta(x^*(J_i), m^*(J_i), J_i)$ or for $m \neq m^*$ are not required to satisfy C3. The only conditions on these evaluations are the conditions in C2; i.e. that chosing an $m \neq m^*$ and any x, or an x different x^* when $m = m^*$ would lead to a perceived evaluation which is less than that from the optimal policy.

Note also that none of our conditions are formulated in terms of beliefs about either the play or the "types" of opponents. There are two reasons for an applied researcher to find this to be a desirable feature of a set of equilibrium conditions. First, as beliefs are not observed, they can not be directly tested. Second, as we will show presently, it implies that we can compute equilibria without ever explicitly calculating posterior distributions. This latter fact which will greatly simplify our computational problem.

Comment 1. Our definition of Applied Markov Equilibria is closely related to the definition of Self Confirming Equilibria in Fudenburg and Levine (1993a). Self Confirming Equilibria weaken the standard Nash equilibrium conditions by requiring that each player's actions is optimal given the player's belief about opponent's actions but that these beliefs need only be correct along the equilibrium path (so no player observes actions which contradicts his beliefs). Our equilibrium conditions explicitly introduce the evaluations that the agents use to determine their optimal actions. These evaluations, together with the primitives of the problem, allow us to contruct values for play along the equilibrium path. Our equilibrium condition insures that these values are consistent with optimizing behavior on points that are visited infinitely often. Of course data generated from a Self Confirming Equilibrium will also produce a set of evaluations. Moreover those evaluations will generate values that satisfy our AME equilibrium conditions. However there is not a one to one relationship between the two concepts. This because the correct valuations could also be generated by incorrect perceptions on competitors' actions. In particular player i may have incorrect beliefs about the play of players j and k if those beliefs generate consistent values for the game. In addition our consistency requirement C3 is defined only for points in $\mathcal{R}$, and imposes no consistency conditions at points outside of $\mathcal{R}$. Self confirming equilibria requires that players have correct beliefs on oponents' actions along the entire equilibrium path, i.e. also for points on the equilibrium path that are not in $\mathcal{R}$.[8]

Comment 2. We now come back to the sense in which our equilibrium conditions are empirically testable. To determine this we need to specify what information the empirical researcher has at its disposal. At most the

[8] Our setup allows games to start at points outside $\mathcal{R}$ but does no impose any conditions on such points.

empiricist will know the union of the information sets of all players at each period, that is our s_t. To determine what is testable in this case it will be useful to use a distinction introduced by Pakes and McGuire (2001). They partition the points in $\mathcal{R}$ into interior and boundary points. Points in $\mathcal{R}$ at which there are feasible (though inoptimal) strategies which can lead to a point outside of $\mathcal{R}$ are labelled boundary points. Interior points are points that can only transit to other points in $\mathcal{R}$ no matter which of the feasible policies are chosen (equilibrium or not). At interior $s \in \mathcal{R}$ we can test condition C3 for every $\{W(\eta, m|J_i), m \in M, \eta \in \Omega(\eta)\}$ for every J_i . As a result we can test C2 for such points. However we can not test C2, that is for the optimality of strategies, at boundary points. This because we will never have accurate estimates of the $\{W(\cdot)\}$ associated with points outside of the recurrent class and we would have to have these evaluations to test whether observed play is optimal[9]. Of course it is likely that the empiricist will observe less than s_t, perhaps only the publically avaialbe information in each period. Provided the empiricist knows (or has estimated) the primitive parameters, testing would then consist of computing the equilibrium associated with those primitives, and then testing whether the observed probabilities of transition from one public information set to another are consistent with the equilbrium calculations.

2.1 The Restriction to a Finite State Space.

We purposefully restricted ourselves to games where the state space was finite. Without this restriction the applied researcher could not be expected to compute the equilibrium, and agents playing the game could not be expected to learn optimal play. In order to insure finiteness of the state space we assumed that there was: (i) an upper bound to the number of firms simultaneously active, (ii) each firm's physical states (our ω) could only take on a finite set of values, (iii) the discrete action was chosen from a finite feasible set, and (iv) the continuous action is not observed by the agent's opponents and affects the game only through its impact on the transition probabilities of the physical state.

These restrictions would insure a finite state space were this a game of

[9] Though the data generating process will not allow us to do a full test of C2 at boundary points, we could in principle construct tests from just necessary conditions for equilibrium policies at the boundary points rather than test the necessary and sufficient conditions. We do not pursue this line of reasoning further here.

complete information (see Ericson and Pakes, 1995) or a game with incomplete information where the only source of incompleteness is a firm specific state variable which is private information and distributed independently over time (see Bajari, Benkard and Levin, forthcoming, or Pakes Ostrovsky and Berry, forthcoming). Unfortunately these restrictions are not enough to insure that a more general dynamic game with incomplete information generates a finite state space. That is with only these restrictions the variables which will be informationally relevant to any one agent will include the entire history that the agent observes.

There are at least two possible ways to insure finiteness. One is to consider a class of games in which there are equilibrium strategies that will always depend on only a finite number of states. We provide the condition that these games must satisfy below. The second is to restrict the information set exogenously; i.e. that is restrict the information agent's have at their disposal exogneously. An example of the latter is to impose a form of imperfect recall that leads to a finite information set.

In discussing the conditions that a game must satisfy for their to be equilibrium strategies that only depend on a finite number of states, it will be useful to consider the following property.

Periodic Revelation of Information. For any starting period there is a $T^* < \infty$ and a $\tau \leq T^*$ with probability one, such that all agent know all payoff relevant random variables at τ.

.

Proposition 1 *It is possible to restrict equilibrium strategies to be a function of a finite state space if and only if the game has the Periodic Revelation of Information property.*

Proof. We will prove the theorem for a game with two competitors and no discrete controls, as this makes the logic underlying the theorem transparent. Let $x_i^t \equiv \{x_{i,t+\tau}(\mathcal{J}_{i,t+\tau})\}$, for $i = \{1, 2\}$ denote a set of strategies for the two agents from period t onward. Then for $x_{i,t}(\mathcal{J}_{i,t})$ to be an equilibrium strategy it must maximize

$$V(\mathcal{J}_{i,t}|x_{i,t}, x^{t+1}) \equiv$$

$$\sum_{\omega_{i,t}}\Big[\sum_{\omega_{-i,t}}\pi(\omega_{i,t},\omega_{-i,t})p(\omega_{-i,t}|\mathcal{J}_{i,t})+\beta E(\sum_{\tau=1}^{\infty}\pi(\omega_{i,t+\tau},\omega_{-i,t+\tau})\beta^{\tau}|\mathcal{J}_{i,t},x^{t+1},\omega_{i,t})]\Big]p(\omega_{i,t}|x_{i,t},\omega_{-i,t-1}).$$

We first prove that if the the periodic revelation of information property does not hold, then we can not truncate the information set. To prove this we show that if one agent does truncate the best response of the other agent is not to truncate.

If equilibrium strategies are truncated at $t-2$ the only information that $x_{-i,t-1}$ can be based on is $\omega_{-i,t-2}$, that is if strategies are truncated $x_{-i,t-1}=x(\omega_{-i,t-2})$. However

$$V(\mathcal{J}_{-i,t-1}|x_{-i,t-1},x^{t}_{-i})\equiv$$

$$\sum_{\omega_{i,t-1},\omega_{-i,t-1}}\pi(\omega_{-i,t-1},\omega_{i,t-1})p(\omega_{-i,t-1}|x_{-i,t-1},\omega_{-i,t-2})p(\omega_{i,t-1}|x_{i,t-2},\omega_{-i,t-2})p(\omega_{i,t-2},x_{i,t-2}|\mathcal{J}_{-i,t-1})$$

$$+\ \beta\ E\Big[\sum_{\tau=0}^{\infty}\pi(\omega_{i,t+\tau},\omega_{-i,t+\tau})\beta^{\tau}|\mathcal{J}_{-i,t-1},x^{t}_{-i},x_{-i,t-1}\Big].$$

3 An AI Algorithm to compute an AME.

In this section we show that we can construct an Applied Markov Equilibrium (henceforth an AME) by using a reinforcement learning algorithm. So just as in Fudenberg and Levine (1993a), our equilibria can be motivated as the outcome of a learning process.[10] The reinforcement learning algorithm is such that at every stage players have valuations regarding the continuation game and they choose their actions optimally given those values. They then use the realized outcomes of the game to correct their evaluations. Thus the algorithm is constructed such that players choose actions optimally even outside the recurrent class but these choices may be based on wrong evaluations (and therefore the equilibrium conditions cannot be consistently tested for these points). Note also that in our algorithm players are not engaged in intentional experimentation. Players choose actions that are optimal given their evaluations without considering actions that are intended to "improve"

[10] But in our case the learning is about the value of alternative outcomes, while in their case it is about the actions of opponent players. Our model of reinforcement learning is that of passive learning.

their evaluations.[11] However, the algorith can be designed in such a way that it would generate lots of experimentations before it converges to the recurrent class.[12]

The algorithm provided in this section is iterative, and we begin by describing the iterative scheme. The rule for when to stop the iterations consists of a test of whether the equilibrium conditions defined above are satisfied, and we describe the test immediately after presenting the iterative scheme. We note that since our algorithm is a simple reinforcement learning algorithm, an alternative approach would have been to view the algorithm itself as the way players learn the values needed to choose their policies, and justify the output of the algorithm in that way. A reader who subscribes to the latter approach may be less interested in the testing subsection[13]. We conclude this section with a brief discussion of the properties of the algorithm; both its computational properties, and its relationship to various conceptual issues discussed in the economic literature.

3.1 The Iterative Procedure.

Our algorithm approximates the $W \equiv \{W(\eta, m|J); \eta \in \Omega(\eta), m \in M, J \in \mathcal{J}\}$ directly using techniques analogous to those used in the stochastic approximation (or reinforcement learning) literature (see footnote 3). The algorithm is iterative. An iteration, say k, is defined by couple

- its location, say $L^k = (J_1, \ldots J_{n(k)})$, defines the information set of the $n(k)$ agents active at iteration k[14], and
- a set of evaluations, W^k.

So to iterate we must update both L^k and the W^k.

Schematically the updates are done as follows. First the algorithm calculates policies for all agents active at L^k. These policies are chosen to

[11] Fudenberg and Kreps (1994) and Fudenberg and Levine (1993b) for example considered models with active experimentation and studied the role of such experimentation in the convergence of the learning process to the Nash equilibrium.

[12] This is done by providing sufficiently high initial evaluations.

[13] On the other hand, there are several issues that arise were one to take the learning approach seriously, among them; the question of whether (and how) an agent can learn from the experience of other agents, and how much information an agent gains about its value in a particular state from the agent's experience in related states.

[14] Active agents include all incumbents, and in models with entry, the potential entrants.

maximize their value (that is to solve condition C2) given the evaluations in memory, W^k. Then computer generated random draws from the distributions in equations (3), (6) and (7) conditional on those policies and the current state are taken. Those draws are used to update both L^k and W^k.

The location is updated using the updating functions in equations (4) (for the public information) and (5) (for the private information) for each of the active agents. This determines L^{k+1}. Next we update W^k. The k^{th} iteration only updates the components of W associated with L^k. It treats the updated $J_i^{k+1} = (\xi^{k+1}, z_i^{k+1})$ as a random draw on the next location and evaluates that location with W^k in memory using the formula in condition C2.. This evaluation is averaged with the evaluations obtained from the prior random draws at the same (η, m, J_i) to obtain the new estimate for that component of W. We now formalize this procedure and then discuss some of its properties.

Details. The reinforcement learning part of the algorithm consists of an iterative procedure and subroutines for calculating initial values and profits. We begin with the iterative procedure.

Each iteration starts with a location, L^k, and the objects in memory, say $M^k = \{M^k(J) : J \in \mathcal{J}\}$. The elements of L^k specify the information sets of the active agents, and are mutually consistent (the components which are public information have the same values for all agents). The elements of $M^k(J)$ specify the objects in memory at iteration k for information set J. $M^k(J)$ contains

- a counter, $h^k(J)$, which keeps track of the number of times we have visited J prior to iteration k, and if $h^k(J) > 0$ it contains
- $W^k(\eta, m|J)$ for $m \in [0, \ldots, M]$ and $\eta \in [0, \ldots, \Omega(\eta)]$.

If $h^k(J) = 0$ there is nothing in memory at location J. If we require $W(\cdot|J)$ at a J at which $h^k(J) = 0$ we have an initiation procedure which sets $W^k(\eta, m|J_i) = W^0(\eta, m|J_i)$. We come back to a discussion of possible choices for W^0 below.

Policies and Random Draws for Iteration k. For each J_i^k which is a component of L^k call up $W^k(\cdot|J_i)$ from memory and choose $(x^k(J_i), m^k(J_i))$ to

$$max_{m\in M} sup_{x\in X}\left[\sum_{\eta} W^k(\eta, m|J_i^k)p_\eta(\eta|x, m, \omega_i^k)\right].$$

With this $\{x^k(J_i^k), m^k(J_i^k)\}$ use equation (1) to calculate the realization of profits for each active agent at iteration k[15]. Next use the distributions in (3), (6) and (7) conditional on the new policies and the information in memory at L^k, together with a pseudo random number generator, to calculate $\left((\eta_i^{k+1}, \mu_i^{k+1})_{i=1}^{n_k}, \epsilon^{k+1}, \nu^{k+1}\right)$.

Updating. Use $\left((\eta_i^{k+1}, \mu_i^{k+1})_{i=1}^{n_k}, \epsilon^{k+1}, \nu^{k+1}\right)$ and equations (4) and (5) to obtain the updated location of the algorithm

$$L^{k+1} = [J_1^{k+1}, \ldots, J_{n^{k+1}}^{k+1}].$$

To update the W it is helpful to define the value of play after profits and the random draws are realized, i.e. to define

$$V^{k+1}(J_i^k) = \pi(\omega_i^k, \omega_{-i}^k, m_i^k, m_{-i}^k, x_i^k, d^k) \; + \tag{9}$$

$$\beta max_{m\in M} sup_{x\in X}\left[\sum_{\eta} W^k(\eta, m|J_i^{k+1})p_\eta(\eta|x, m, \omega_i^{k+1})\right].$$

Note that the equilibrium W is just the expectation of $V(\cdot)$ given the information available prior to the realizations of profits and the other random variables. Consequently we update W as we do an average, i.e. we set

$$W^{k+1}(\eta, m|J_i^k) - W^k(\eta, m|J_i^k) = \frac{1}{A\left(h^k(J_i^k)\right)}[V^{k+1}(\hat{J}_i^k) - W^k(\eta, m|J_i^k)], \tag{10}$$

where $A(\cdot) : \mathcal{Z}^+ \to Z^+$, is increasing, and satisfies Robbins and Monroe's conditions (1956)[16]. Setting $A(h^k(J_i)) = h^k(J_i)+1$, the number of times point J_i^k had been visited by iteration $k+1$ would do, and produces an estimate of

[15] If d is random, then the algorithm has to take a random draw on it before calculating profits.

[16] Those condition are that the sum of the weights of each point visited infinitely often must increase without bound while the sum of the weights squarred must remain bounded.

$W^k(J_i^k)$ which is the simple average of the $V^r(J_i^r)$ over the iterations at which $J_i^r = J_i^k$. However since the early values of $V^j(\cdot)$ are typically estimated with more error than the later values, it is often useful to give them lesser weight. We come back to this point below.

That concludes an iteration.

3.2 Testing For an Equilibrium.

This subsection assumes we have a W vector which is outputted at some iteration of the algorithm, say $W = \tilde{W}$, and provides a test of whether that vector generates AME policies and values on a subset $\mathcal{R}(\tilde{W}) \in \mathcal{S}$.

Once we substitute $\tilde{W}$ into condition C2 we determines policies for all agents active at each $s \in \mathcal{S}$. These policies determine the probabilities of transiting to any future state. Let the probability of transiting from s to s' be denoted by $q(s', s|\tilde{W})$, where $0 \leq q(s', s|\tilde{W}) \leq 1$, and $\sum_{s' \in \mathcal{S}} q(s', s|\tilde{W}) = 1$. Now order the states and arrange these probabilities into a row vector in that order, say $q(s|\tilde{W})$. Do this for each $s \in \mathcal{S}$, and combine the resultant rows into a matrix whose rows are ordered by the same order used to order the elements in each row. The result is a Markov matrix (or transition kernel) for the industry structures, say $Q(\cdot, \cdot|\tilde{W})$. This matrix defines the Markov process for industry structures generated by $\tilde{W}$. $Q(\cdot, \cdot|\tilde{W})$ is a finite state kernel and so any sample path generated by it will, with probability one in a finite number of perids, enter a recurrent class, say $\mathcal{R}(\tilde{W}) \subset \mathcal{S}$, and once within it will stay within it forever.

To test whether $\tilde{W}$ generates a process which satisfies the conditions defining an AME we first need a candidate recurrent class of $Q(\cdot, \cdot|\tilde{W})$. To obtain our candidate for $\mathcal{R}(\tilde{W})$, we start at any s^0 and use $Q(\cdot, \cdot|\tilde{W})$ to simulate a sample path $\{s^j\}_{j=1}^{J_1+J_2}$. Let $\mathcal{R}(J_1, J_2)$ be the set of states visited at least once between $j = J_1$ and $j = J_2$, and $\mathcal{P}(\mathcal{R}(J_1, J_2))$ be the empirical measure of how many times each of these states was visited. Then, as both J_1 and $J_2 - J_1 \to \infty$, $R(J_1, J_2)$ must converge to a recurrent class of the the process $Q(\cdot, \cdot|\tilde{W})$, and hence satisfies our condition C1, and $\mathcal{P}(\mathcal{R}(J_1, J_2))$ converges to an invariant measure which provides the limiting frequency of visits to these states. As we shall see it typically does not take long to generate a million iterations of the stochastic algorithm. As a result it is easy to simulate several million draws, throw out a few million, and then consider the locations visited by the remainder as the recurrent class.

Note that we have constructed $Q(\cdot, \cdot|\tilde{W})$ in a way that insures that condi-

tion $C2$ is satisfied everywhere. So what remains is to test whether condition $C3$ is satisfied at every $s \in \mathcal{R}(J_1, J_2)$. One way to construct a test of this condition is to compute the integrals on the right hand side of the conditions defining the equilbrium W in C3 using the policies generated by $\tilde{W}$, and then base a test statistic on the difference between the computed values and $\tilde{W}$. This would be analogous to the test used by Pakes and McGuire (2001). As noted their that test is subject to a curse of dimensionality, and even for moderately sized problem the computational burden of the test can greatly exceed the burden of the iterations leading to the test. As a result we provide a test of condition C3 which is not subject to the curse, but still has an interpretation as the extent of approximation error in our estimates. In particular, instead of computing the right hand side of the integrals in condition C3 we approximate them using simulation (which avoids the curse of dimensionality) and then account for the simulation error.

So our test consists of measuring the difference between the estimates of $\tilde{W}(\eta, m^*|J_i)$ in memory, and an approximation to the expected discounted values of future net cash flows that an agent with the information set J_i and the random draw η_i would obtain were all agents using the policies generated by $\tilde{W}$. The approximation is a sample average of the discounted value of net cash flows from simulated sample paths starting at (J_i, η_i). The squared differences between the $\tilde{W}$ and the average of the discounted value over the simulated sample paths is a sum of; (i) the sampling variance in the average of the discounted value of the simulated sample paths (we will refer to this as the sampling variance term), and (ii) the difference between the *expectation* of the discounted net cash flows from the simulated paths and required components of $\tilde{W}$ (we will refer to this as the "bias" term). We subtract a consistent estimate of the sampling variance term from this squared difference to obtain a test statistic which, at least in the limit, will depend only on the bias term.

The test is constructed as follows. Start at an initial $s^0 \in \mathcal{R}$ and an initial draw on η for each J_i component of s^0, i.e. at a set of couples $\{(J_i^0, \eta_i^0)\}_{i=1}^{n^0}$, where n^0 is the number of active agents at s^0. Now simulate draws for $\left((\eta_i^1, \mu_i^1)_{i=1}^{n^0}, \epsilon^1\right)$ using the policies generated by $\tilde{W}$. Use these simulation draws to compute

$$\hat{W}^{l=0}(\eta, m^*(J_i^0)|J_i^0) \equiv \pi(J_i^0, J_{-i}^0, m^*(J_i^0), m^*(J_{-i}^0), x^*(J_i^0), d^0) + \beta W(\eta_i^1, m^*(J_i^1)|J_i^1),$$

where

$$J_i^1 = (\xi^1, z_i^1), \quad \xi^1 = G_\xi(\xi^0, \nu^0, \epsilon^1), \quad z_i^1 = G_z(\xi^0, z_i^0, \mu_i^1),$$

for each of the n^0 points (J_i^0, η_i^0). Then keep in memory at a location which is (J_i, η_i) specific; (i) $\hat{W}^0(\eta, m^*(J_i^0)|J_i^0)$, (ii) the square of this, say $S\hat{W}^0(\eta, m^*(J_i^0)|J_i^0) = \hat{W}^0(\eta, m^*(J_i^0)|J_i^0)^2$, and (iii) an initialized counter, say $h^{l=0}(J_i, \eta_i) = 1$.

Now consider the simulated locations $\{(J_i^1, \eta_i^1)\}_{i=1}^{n^1}$. At each of these points simulate as above and compute

$$\hat{W}^1(\eta, m^*(J_i^1)|J_i^1) \equiv \pi(J_i^1, J_{-i}^1, m^*(J_i^1), m^*(J_{-i}^1), x^*(J_i^1), d^1) + \beta W(\eta_i^2, m^*(J_i^2)|J_i^2).$$

If (J_i^1, η_i^1) is the same as one of the values (J_i^0, η_i^0), average the two values of $\hat{W}(\cdot)$ and $S\hat{W}(\cdot)$ at that location, call the averages $A\hat{W}^1(\cdot)$ and $A\hat{S}W^1(\cdot)$, and keep them together with a value for $h^l(\cdot)$ equal to 2, in memory at that location. If a particular (J_i^1, η_i^1) was not visited prior to this start a new location, set

$$\hat{W}^1(\eta, m^*(J_i^1)|J_i^1) = \pi(J_i^1, J_{-i}^1, m^*(J_i^1), m^*(J_{-i}^1), x^*(J_i^1), d^1) + \beta W(\eta_i^2, m^*(J_i^2)|J_i^2),$$

$S\hat{W}^1(\cdot)$ accordingly, and $h^1(J_i^1, \eta_i^1) = 1$. We continue in this manner until a large number of periods are simulated.

If we let E be the expectation operator over the simulated random draws then

$$E\left(\frac{A\hat{W}(\eta_i, m^*(J_i)|J_i)}{\tilde{W}(\eta_i, m^*(J_i)|J_i)} - 1\right)^2 = E\left(\frac{A\hat{W}(\eta_i, m^*(J_i)|J_i) - E\left[A\hat{W}(\eta_i, m^*(J_i)|J_i)\right]}{\tilde{W}(\eta_i, m^*(J_i)|J_i)}\right)^2 + \left(\frac{E\left[A\hat{W}(\eta_i, m^*(J_i)|J_i)\right] - \tilde{W}(\eta_i, m^*(J_i)|J_i)}{\tilde{W}(\eta_i, m^*(J_i)|J_i)}\right)^2. \tag{11}$$

The first term after the equality in (11) is the sampling variance, and the second term is the bias.

Moreover if we let

$$\hat{Var}(A\hat{W}(\eta_i, m^*(J_i)|J_i)) \equiv A\hat{S}W(\eta_i, m^*(J_i)|J_i)\frac{h(\eta_i, J_i)}{h(\eta_i, J_i) - 1} + A\hat{W}(\eta_i, m^*(J_i)|J_i), \tag{12}$$

then

$$E\Big[\hat{Var}(A\hat{W}(\eta_i, m^*(J_i)|J_i))\Big] = E\left(\frac{A\hat{W}(\eta_i, m^*(J_i)|J_i) - E\Big[A\hat{W}(\eta_i, m^*(J_i)|J_i)\Big]}{\tilde{W}(\eta_i, m^*(J_i)|J_i)}\right)^2,$$

and we have an unbiased estimate of the sampling variance. Consequently if

$$\hat{Bias}(AW(\eta_i, m^*(J_i)|J_i))^2 \equiv \Big(\frac{A\hat{W}(\eta_i, m^*(J_i)|J_i)}{\tilde{W}(\eta_i, m^*(J_i)|J_i)} - 1\Big) - \hat{Var}(A\hat{W}(\eta_i, m^*(J_i)|J_i)), \tag{13}$$

then $\hat{Bias}(AW(\eta_i, m^*(J_i)|J_i))^2$ is an unbiased estimate of the square of the percentage bias in $A\hat{W}(\eta_i, m^*(J_i)|J_i))$. Since higher order moments of this estimate are finite, any weighted average of independent estimates of the bias terms over the recurrent class of points will converge to the same weighted average of the true bias term across these points (a.s.).

The test statistic, to be denoted by $\mathcal{T}$, is an $L^2(\mathcal{P}(\mathcal{R}(J_1, J_2))$ in the bias terms defined in equation (13). More formally

$$\mathcal{T} \equiv \left\| n_s^{-1} \sum_{i=1}^{n_s} \sum_{\eta_i} \frac{h(\eta_i, J_i)}{\sum_{\eta_i} h(\eta_i, J_i)} \hat{Bias}(AW(\eta_i, m^*(J_i)|J_i))^2 \right\|_{L^2(\mathcal{P}(\mathcal{R}(J_1, J_2))}.$$

Assuming the computer's calculations are exact, $\mathcal{T}$ will tend to zero as the number of simulation draws used in the test grows large if and only if $\tilde{W}$ satisfies condition C3. More generally $\mathcal{T}$ is a consistent estimate of the average percentage difference between the two sides of that fixed point in C3. We assume we are "at an equilibrium" when it is sufficiently small[17].

[17]Note that a consistent estimate of the variance of this statistic can be obtained by running this procedure many times and calculating the variance in our statistic over these runs. To use this to produce a traditional one-sided statistical test we would need to; (i) decide what is an acceptable percentage error (if for no other reason then to allow for the imprecision of the computer's calculations) and (ii) decide on the size of the test (the probability of type I error we are willing to accept). The size issue is complicated by the fact that by increasing the number of simulation draws we are free to increase the power of any given alternative to one. I.e. before we proceeded in this way we would want to formalize tradeoff between size, power, and the number of simulation draws.

3.3 Properties of the Algorithm.

First a note on the computational burden of the algorithm. As noted the algorithm does not generate a curse of dimensionality. Perhaps more important in the context of dynamic games with assymetric information, just as the formulation of our equilibrium concept does not require us to formalize posterior distribution functions, our test does not require us to compute posterior distributions. An equilibrium concept and algorithm which required posterior beliefs about player's "types" would increase the computational burden of the algorithm substantially.

We note that, as is the case for all known algorithms designed to compute equilibria for (nonzero sum) dynamic games, there is no guarantee that our algorithm will converge to equilibrium values and policies; that is all we can do is test whether the algorithm outputs equilibrium values, we can never guarantee convergence to an equilibrium *a priori*. Moreover there is no guarantee that the equilibrium is unique. Further, as in the case of any algorithm which needs to select among multiple equilibria, the way we initiate the algorithm, i.e. our choice for W^0, is likely to play a role in this selection.

In particular low initial values are likely to discourage experimentation. As noted by Fudenberg and Kreps (1994) a learning process can converge to a non-Nash outcome when players engage in an insufficient amount of experimentation. In our context we would see this by low values for outcomes which could be reached from boundary points if feasible but inoptimal policies were followed. A way to insure experimentation is to chose high initial values. On the other hand high initial values will tend to result in a longer computational times and a need for more memory. High initial values tend to make the algorithm do more exploration of policies that lead to points outside of the recurrent class, and longer times to convergence on the recurrent class. So there is a tradeoff between experimentation and computational time in selecting among alternative W^0.

There are other aspects of the computational burden that can be varied as well. Our test insures that the $\tilde{W}$ outputted by the algorithm is consistent with the distribution of current profits and the discounted evalutions of the next period's state. We could have considered a test based on the distribution of discounted profits over τ periods and the discounted evaluation of states reached in the τ^{th} period. We chose $\tau = 1$ because it generates the stochastic analogue of the test traditionally used in iterative procedures to determine whether we have converged to a fixed point. It may well be that a different

τ provides a more discerning test, and with our testing algorithm it is not computational burdensome to increase τ.

Finaly since our estimates of the $\tilde{W}$ are formed as sample averages, we expect the estimates from a particular location to be more accurate the more times we visit that location (the larger $h(\cdot)$). If one is particularly interested in policies and values at a given point, for example at a point that is consistent with the current data on a given industry, one can increase the accuracy of the relevant estimates by restarting the algorithm repeatedly at that point.

4 Example: Maintenance Decisions in An Electricity Market.

Market design has been a major issue in the study of markets for generating electricity. Most of the detailed structural work on this topic has ignored the question of when to to take down the plant and do maintenance (check with Catherine). This in spite of the fact that maintenance decisions can have significant impacts on market price, and more generally on the efficiency, of the design.

Here we provide the outlines of a model that can endogenize the maintenance decision. In the model the level of costs of a generator evolve on a discrete space in a nondecreasing random way until a maintenance decision is made. We assume that a firm knows the cost position of its own generators, but not of those of its competitors. This is the source of assymetric information.

Firms can hold their generators off the market for a single period and do maintenance. Whether they do or do not do maintenance is public information. If they do maintenance the cost level of the generator reverts to a base state (to be designated as the zero state) in the next period. If they do not do maintenance they bid a suppy function and compete in the market. In periods when a generator is bid the level of its costs deterorates stochastically.

Initially we assume that if a firm submits a bid function for producing electricity from a given generator, it always submits the same bid. However we will allow heterogeneity among both cost and bidding functions across agents. We then provide a second example where there is some discretion over the submitted bid function.

In the initial case a firm's competitors can not infer anything from the bid except that the firm did bid. However the distribution of a firm's competitor's states depends on how many periods passed since the competitor did maintenance. Moreover the competitor will determine when to do maintenance as a function of the amount of time since *its* competitors have done maintenance. Consequently the number of periods since the last bid of each generator (a firm's own, as well as its competitor's states) contain information on the likelihood of future maintenance decisions, and so are informationally relevant and become state variables in the problem. We also allow demand to vary exogenously with the day of the week. Consequently the "payoff relevant" state variables of a firm are the level of costs of its generators and the day of the week, and the "informationally relevant" state variables of the firm will be the number of periods since the last maintenance period of each generator in the system, a vector which is public information. If $\tau_{i,j}$ is the amount of periods since maintenance was done on the i^{th} generator of the j^{th} firm, the state space of this problem will be finite provided that maintenance must be done at least once in every $\overline{\tau}$ periods. Alternatively the state space will be finite if there is an ω state at which maintenance must be performed and that state is reached in a finite number of periods with probability arbitrarily close to one.

4.1 The Model.

We let $\omega_{j,i,t} \in \Omega = \{0, 1, \ldots, \overline{\omega}\}$ be the cost state of generator j of firm i in period t. State "0" occurs just after maintenance and is the lowest cost state. At state $\overline{\omega}$ one simply cannot use the generator. Each firm owns multiple generators and we let $\omega_{i,t} = \{\omega_{i,j,t}; j = 1 \ldots n_i\} \in \Omega^{n_i}$. The cost function for producing electricity differs by generator and is given by $c_{i,j}(\cdot,\cdot) : Y \times \Omega \to R_+$ where $y \in Y$ indexes possible output levels. We assume this function is nondecreasing in both its arguments.

The cost state of a generator evolves as

$$\omega_{j,i,t+1} = \omega_{j,i,,t} + \eta_{j,i,,t},$$

where $\eta_{j,i,t} = -\omega_{j,i,t}$ if there is maintenance (in which case $\omega_{j,i,t+1} = 0$), and has distribution $P_\eta(\cdot)$ with support on the positive integers otherwise.

$\tau_{j,i,t} \in \mathcal{T}_{j,i} \equiv \{0, \ldots \overline{\tau}_{i,j}\}$ is the number of periods since the last maintenance for generator j of firm i, with $\tau_{i,t} = [\tau_{1,i,t} \ldots \tau_{n_i,i,t}] \in \Pi_j \mathcal{T}^{j,i} \equiv \mathcal{T}^{n_1}$.

Note that we have assumed that there is a maximum number of periods (our $\overline{\tau}_{i,j}$) after which the generator must be maintained.

We will have two firms competing in this market. Let $J_{i,t}$ be the information set of agent i in period t (we come back to listing to what is included in this below). The strategy of firm i is then a choice of

$$m_i = [m_{1,i}, \ldots m_{n_i,i}] : \mathcal{J} \to \Pi_j \Big(0, m_{j,i}(\cdot)\Big) \equiv M_i,$$

where $m_{j,i}(\cdot) : Y \to R_+$ is a bid function. That is a firm either bids generator j with a given bid function $m_{j,i}(\cdot)$, or takes that generator down to do maintenance (in which case $m_{j,i,t} = 0$). We assume that whenever the firm witholds a generator from the market they do maintenance on that generator, and that the cost of this maintenance is $cm_{j,i,t} \in R_+$. Note also that these definitions imply that

$$\tau_{i,t+1} = I\{m_{i,j,t} > 0\} \times [\tau_{i,t} + 1] + I\{m_{i,j,t} = 0\} \times 0,$$

where here and below $I\{\cdot\}$ is the indicator function which takes the value of one when the $\cdot$ condition is satisfied and zero elsewhere.

Demand for electricity will be given by $D(\cdot, \cdot) : d \times p \to R_+$ where d indexes the day of the week and p indexes price. We let $f : M^{n_1} \times M^{n_2} \times D \to Y^b \times Y^s \times P$, be an allocation rule (or a market design) which takes strategies and demand as input and generates output allocations for each firm and a price. We assume that the price paid per unit of electricity does not differ over generators or between firms.

The profit function is then is defined as $\pi_i : M^{n_i} \times M^{n_{-i}} \times \Omega^{n_i} \times D \to \mathcal{R}_+$ is the profit function

$$\pi_i\Big(m_{1,t}, m_{2,t}, d_t, \omega_{i,t}\Big) = p(m_{1,t}, m_{2,t}, d_t) \sum_j y_{i,j,t}(m_{1,t}, m_{2,t}, d_t)$$

$$- \sum_j \Big[I\{m_{i,j,t} > 0\} c(\omega_{i,j,t}, y_{i,j,t}(m_{1,t}, m_{2,t}, d_t)) - I\{m_{i,j,t} = 0\} cm_{j,i} \Big].$$

This is a special case of our general model. In particular there are two firms (with no entry or exit), and no x or investment. Also ω_{-i} does not enter the profits of firm i so, given the observable m_{-i}, profits per se do not contain any information about the state of the generators of the other firms.

As a result the only private information is ω_i (or in our previous notation $z_i = \omega_i$) and the only new information on ω_i is η_i (so $\mu_i = \eta_i$).

The public information is

$$\xi_t = (\tau_{1,t}, \tau_{2,t}, d) \in \mathcal{T}^{n_1} \times \mathcal{T}^{n_2} \times D.$$

Since private information for firm i is $\omega_{i,t}$, the firm's information set is

$$J_{i,t} = (\omega_{i,t}, \xi_t), \text{ and } J_{i,t} \in \mathcal{J}_i,$$

and

$$s_t = (J_{1,t}, J_{2,t}) \in \mathcal{S}.$$

Given this specification for the primitives, the AMPE equilibrium is defined as in Section 2, and the algorithm used to compute and test for the equilibrium is as described in Section 3.

4.1.1 Parameterization for Numerical Analysis.

- Two firms ($n = 2$). One firm employs large generators and the other employs small generators. Larger firms have larger capacity, lower marginal costs, larger $\overline{\tau}$, and larger maintenance costs.
- n_s generators for the small firm and n_b generators for the firm with the small generators.
- cost of maintenance is cost per day $\times$ capacity.
- take $\Omega = (0, 1, \ldots, 5)$ and at $\omega = 5$ no one can produce, costs go up by 10% with each ω incarese.
- bids: highest cost.
- d is day of the week,
- Demand see below. Moreover we want it to be able to shift between the high bid and the low bid. So some demand at the bid of the nuclear should be less than $n * K/2$ times the capcity of a nuclear generator. Some demand at price equal to the high cost generator must be greater than total capacity.

- probabilities of going from one ω to the next is about 20% and the probabilities are operative whenever there is no maintenance (regardless of production),

Demand is log linear;

$$log(Q) = D_d - \alpha log(P),$$

where D_d differs fro weekday and weekend and $\alpha = 1.2$.

The supply function $q^s(p)$ given our bidding function is as follows

$$q^s(p) = \begin{cases} 0 & p < 10 \\ 25M_b & p = 10 \\ 25M_b + M_b(\frac{p}{10} - 1) & 10 < p < 20 \\ 26M_b + 15M_s & p = 20 \\ 25M_b + M_b(\frac{p}{10} - 1) + 15M_s + M_s(\frac{p}{10} - 2) & p > 20 \end{cases}$$

4.1.2 Alternative Market Designs.

We will look at a number of alternatives.

- Duopoly, as above.
- Social planner that controls m, observes the ω, and bidding functions are determinned as in standard duopoly. See the discussion below for details.
- Social planner knows only τ.
- Monopoly no regulaton but constrained to have standard bidding function (computation as with social planner but with different objective function).
- Monopoly regulated prices. Monopoly must satisfy demand at a fixed price. It minimizes cost. We can chose the price and look for the price that maximizes social welfare.
- Regulated Duopoly, where the bids submitted are dependent on τ in the following manner. The cost function bid at $\tau = j$ is the cost function when $\omega = j$ for $0 = j \leq 3$. For $\tau > 3$ we use the cost function at $\omega = 3$.

Social Planner. Let J_t^{sp} be the information set of the planner in period t. This includes the τ's and the $\omega' s$ of both firms, or

$$J_t^{sp} \equiv (d, \omega^s, \tau^s, \omega^b, \tau^b).$$

The strategy of the social planner is given by

$$m = [m^s, m^b] : \mathcal{J}^{sp} \rightarrow \{0,1\}^{n_s} \times \{0,1\}^{n_b},$$

that is what the social planner does is decide which of the generators to bid in every period. The rules governing the evolution of the state variables and the demand function are as before.

The market maker has an allocation rule which allocates output to the two firms and hence sets price based on the information it has at its disposal (which now includes the states of all generators, the generators bid in, and the day of the week). We let that allocation rule be

$$f^{sp} : J^{sp} \times m \times d \rightarrow (Y^b, Y^s, p).$$

We will do this allocation as before. The firm's submit the same bid functions, we then horizontally sum these curves and intersect the result with demand.

The planner the choses m to maximize the expected discounted value of the sum or consumer surplus and producer profits. More formally the planner's one period return function is

$$\pi^{sp}(\mathcal{J}^{sp}, m) =$$

$$CS(p,d)+\pi^b(\omega^b, Y^b, m^b, p)+\pi^s(\omega^s, Y^s, m^s, p)-\sum_j(1-m_j^b)cm^b-\sum_j(1-m_j^s)cm^s,$$

where

$$CS(p,d) \equiv \int_p^\infty D(x,d)\,dx$$

and

$$\pi^i(\omega^i, Y^i, m^i, p) \equiv p\times Y^i - Min_{y_{i,1},\ldots y_{i,n_i}}\{\sum_j C(y_{i,j}m_{i,j}, \omega_{i,j}) : \sum_j y_{i,j}m_{i,j} = Y_i\}.$$

The planner's value function is then

$$V^{sp}(\mathcal{J}^{sp}) \equiv max_m\{\pi^{sp}(\mathcal{J}^{sp}, m) + \sum_{\mathcal{J}'^{,sp}} V^{sp}(\mathcal{J}'^{,sp})pr(\mathcal{J}'^{,sp}|\mathcal{J}^{sp}, m)\}.$$

Alternatively we can define

$$W^{sp}(m|\mathcal{J}^{sp}) \equiv \pi^{sp}(\mathcal{J}^{sp}, m) + \sum_{\mathcal{J}'^{,sp}} V^{sp}(\mathcal{J}'^{,sp}) pr(\mathcal{J}'^{,sp}|\mathcal{J}^{sp}, m)$$

and the planner's problem is

$$\max_m W^{sp}(m|\mathcal{J}^{sp}).$$

Computation. Similar to before. We require starting values. Suggestion is to sum the values of the game to the two agents when $\mathcal{J}^b \cup \mathcal{J}^s = \mathcal{J}^{sp}$.

Non-regulated Monopoly. Same procedure just drop consumer surplus from the $\pi^{sp}(\cdot)$ function.

Regulated Monopoly. The regulation will be in the form of a price. The problem will be to find the price that optimize social surplus.

Given a price vector, say p, we get a quantity demanded of the monopolist. The monopolist then choses how to supply that quantity by chosing which generators operate and how to allocate the quantity among the operating generators.

The only difference between the regulated and the non-regulated monopolist is in the profit function. The regulated monopolist must supply all the output demanded at that price so the demand function and price gives us the outputs (Y_w, Y_d). The profit for the regulated monopolist conditional on m is then

$$\pi(p, m) \equiv pY(w) - min_{y_{1,s},\ldots y_{n_s,s}, y_{1,b},\ldots y_{n_b,b}} \{\sum_i \sum_j C(y_{i,j} m_{i,j}, \omega_{i,j}) : \sum_i \sum_j y_{i,j} m_{i,j} = Y_w\}.$$

Regulated Duopoly. Here the only thing different between this an duopoly is the bids that are submitted are dependent on τ in the following manner. The cost function bid at $\tau = j$ is the cost function when $\omega = j$ for $0 = j \leq 3$. For $\tau > 3$ we use the cost function at $\omega = 3$. Everything else proceeds as in duopoly. Now the difference is the supply function depends on τ.

Dimension of state space. Since there is symmetry the dimension of the (τ_s, τ_b) couples is approximately

$$\frac{\overline{\tau_s}^{n_s}}{n_s!}\frac{\overline{\tau_b}^{n_b}}{n_b!}$$

To see this note that the number of orderred ω tuples is

$$\sum_{\tau_1=0}^{\overline{\tau}_s}\sum_{\tau_2=0}^{\tau_1}\sum_{\tau_3=0}^{\tau_2}\cdots\sum_{\tau_s=0}^{\tau_{s-1}} 1$$

and use the formula that $\sum_{j=0}^{j_1} 1 \approx j_1^2/2$. This implies that we are limited in the size of $\overline{\tau}$.

Similarly the dimension of the ω_i space will be approximately $6^{n_s}/n_s!$ and $6^{n_b}/n_b!$, while the dimension of the d space is 7.

So

$$\#\mathcal{J}_s \approx 6^{n_s}/n_s! \times \frac{\overline{\tau_s}^{n_s}}{n_s!}\frac{\overline{\tau_b}^{n_b}}{n_b!} \times 7$$

and there is a similar number for the firm that employs big generators. Of course the recurrent class may be much smaller than this.

4.1.3 Memory and Updates

Let k be an index of the iteration number, and $h^k(J_i)$ be the number of times location J_i was visited prior to iteration k.

Memory If $h^k(J_i) > 0$ we have in memory $(h^k(J_i), \pi^{k,E}(J_i, m_i), W^k(\eta_i, m_i|J_i))$ for each $J_i \in \mathcal{J}$ and $m_i \in M_i$. Note that in the simple case with only one strategy per generator, $(\pi^{k,E}(J_i, m_i)$, is a vector of dimension at most 2^{n_i}. $W^k(\eta_i, m_i|J_i))$ can be as large as 3^{n_i} (the alternatives are; $(m_i = 0, \eta_i = -\omega_i)$, $(m_i = 1, \eta_i = 0)$ and $(m_i = 1, \eta_i = 1)$). Symmetry considerations actually make both these vectors somewhat smaller.

If $h^k(J_i) = 0$ nothing is in memory for that point at that iteration.

Updates We need to update $\pi^E(m_i, J_i)$ and $W(\eta_i, m_i|J_i)$ for each firm at each iteration. The updates are done only for point J_i^k, for $i = 1, 2$, that is only for the information set visited at iteration k. They are done after the optimal policies are chosen and the η are drawn for that iteration. I will call the draws on the η, η^{k+1} and the policies $m^k(\cdot)$.

We first consider a point where $h^k(J_i) > 0$. Then the update for $\pi(\cdot)$ which needs to be done for $m_i = \{0, 1\}$ for each combination of distinct ω_i (so if there are two generators at the same ω we need only do it three times, as above....)is

$$\pi^{E,k+1}(m_i, J_i^k) = \frac{h^k(J_i^k)}{h^k(J_i^k)+1}\pi^{E,k}(m_i, J_i^k) + \frac{1}{h^k(J_i^k)+1}\pi_i(m_i, m_{-i}^k, d^k, \omega_i^k),$$

and the update for $W(\cdot)$ which must be done for the (m_i, τ_i) combinations listed above is

$$W^{k+1}(\eta_i, m_i|J_i^k) = \frac{h^k(J_i^k)}{h^k(J_i^k)+1}W^k(\eta_i, m_i|J_i^k) + \frac{1}{h^k(J_i^k)+1}V(\omega_i^k+\eta_i, \tau^{k+1}(m_i, \tau_i^k), \tau_{-i}^{k+1}, d^{k+1}|W^k)$$

where

$$V(J_i^k|W^k) = \max_{m_i \in M^{n_1}} \Big[\pi_i^{k,E}(J_i^k, m_i) + \beta \sum_{\eta_i} W^k(\eta_i, m_i|J_i^k)p(\eta_i|m_i)\Big].$$

If $h^k(J_i) = 0$ then use the following initiation values.

$$\pi^{E,0}(m_i, J_i) = \pi_i(m_i, m_{-i} = 0, d^k, \omega_i^k).$$

and,

$$W^0(\eta_i, m_i|J_i) = \frac{\pi_i(m_i, m_{-i} = 0, d^k, \omega_i^k + \eta_i(m_i))}{1-\beta},$$

where $\eta_i(m_i) = -\omega_i^k$ if $m_i = 0$ and $\eta_i(m_i)$ can be either 0 or 1 if $m_i = 1$. I.e. there are three components of $W^k(\cdot)$ associated with each J_i.

4.1.4 Initiating the Algorithm, and Iteration Steps.

Start at a particular $J_T^0 = (\omega_1, \omega_2, \tau_1, \tau_2, d)$. This defines (J_1^0, J_2^0).

Note that probably all the J_T with the same (τ_1, τ_2, d) should be kept near each other in memory, as then we can first search for (τ_1, τ_2, d) and then find the (J_1, J_2) that go with it.

For each J_i^0 (for $i = 1, 2$) compute m_i as the solution to

$$\max_{m_i \in M(J_i)} \Big[\pi_i^{E,0}(m_i, J_i^0) + \beta \sum_{\eta_i} W^0(\eta_i, m_i|J_i^0)p(\eta_i|m_i)\Big].$$

where $M(J_i) = \Pi_{j=1}^{n_i}\Big[\chi(0 < \tau_{i,j} < \overline{\tau})[0,1] + \chi(\tau_{i,j} = 0)[1] + \chi(\tau_{i,j} = \overline{\tau})[0]\Big]$, and $\chi(\cdot)$ is the indicator function which takes the value of one if $(\cdot)$ is satisfied and zero elsewhere. That is if a generator is at $\tau = 0$ it can ony chose $m = 1$, if it is at $\tau = \overline{\tau}$ it can only chose $m = 0$ and otherwise it can chose 0 or 1.

This produces (m_1^0, m_2^0). To proceed we also need to draw (η_i^0, η_{-i}^0) for all generators (whether they are turned on by (m_1^0, m_2^0) or not). We also define $\eta_i^0(m_i) = \eta_i^0$ if $m_i = 1$ and $\eta_i^0(m_i) = -\omega_i^0$ if $m_i = 0$

Update Location. The location is updated as follows.

$$J_i^1 = (\tau_1^1 = \tau(\tau_1^0, m_1^0), \tau_2^1 = \tau(\tau_2^0, m_2^0), d^1 = d^0 + 1, \omega_i^1 = \omega_i^0 + \eta_i^0(m_i))$$

for $i = (1, 2)$. J_T^1 is constructed accrodingly.

Update $\pi^E(m_i, J_i^0)$ and $W(\eta_i, m_i|J_i^0)$. They are updated as follows. For $i = 1, 2$ and each m_i update

$$\pi^{E,1}(m_i, J_i^0) = \frac{1}{2}\pi^{E,0}(m_i, J_i^0) + \frac{1}{2}\pi_i(m_i, m_{-i}^0, d^0, \omega_i^0).$$

while for $i = (1, 2)$ and each feasible (η_i, m_i) update

$$W^1(\eta_i, m_i|J_i^0) = \frac{1}{2}W^0(\eta_i, m_i|J_i^0) + \frac{1}{2}V(J_i = (\tau_i^1, \tau_{-i}^1, d^1, \omega_i^0 + \eta_i^0(m_i))|W^0).$$

Storage. We now have to store $\{\pi^{E,1}(m_i, J_i^0)\}_{m_i,i}$,$\{W^1(\eta_i, m_i|J_i^0)\}_{\eta_i,m_i,i}$ and $h(J_i) = 1$ in memory at location J_i for $i = 1, 2$.

Note that probably we should order the memory first by (τ_1, τ_2, d) and then by (ω_1, ω_2), as that will insure we get the tuples together. Still their is a real question of how to do the storage.

Continue. We then continue with the iterative process from $J_T^1 = (J_1^1, J_2^1)$, calling up things from memory as needed.

4.2 Model 2: Further Examples.

4.2.1 Athey and Bagwell.

As another example we use Athey and Bagwell (2004), hereafter AB, and present the modifications that need to be done in their model so it can fit

out structure. AB consider a dynamic price game in which prices are publicly observed while cost position are subject to privately observed persistent cost shock.. The focus of AB is the optimal collusive scheme for such an industry. The firms cost position in period t is either high or low and given by $\omega_{i,t} \in \Omega = \{\omega^h, \omega^l\}$ and change over time according to a first order Markov process such that the probability distribution of $\omega_{i,t+1}$ is given by $F(\omega_{i,t+1}|\omega_{i,t})$.[18] In each period the firms make an announcement regarding its cost position, this announcement can be captured by our discrete control $m_{i,t} \in \{\omega^h, \omega^l\}$. After observing the announcement of all the firms each firm choose its price $p_{i,t} \in R_+$ and the maximum market share it is willing to sell, $q_{i,t} \in [0,1]$. Lets denote these strategies as $x_{i,t} \equiv (p_{i,t}, q_{i,t}) \in R_+ \times [0,1]$. The per period profits is then given by $\pi(\omega_{i,t}, x_{i,t}, x_{-i,t})$.

The public information in this model is the observable public history which is the sequence of realized reports, prices and market share restrictions, $\xi_t = \{m^t, x^t\}$ where $m^t = \{(m_{i,1}, m_{-i,1}), ..., (m_{i,t-1}, m_{-i,t-1})\}$ and $x^t = \{(x_{i,1}, x_{-i,1}), ..., (x_{i,t-1}, x_{-i,t-1})\}$. The private information is the firm's cost position and thus $z_{i,t} = \omega_{i,t}$ and thus $J_{i,t} = (\xi_t, z_{i,t}) \in \mathcal{J}_i$.[19] The timeline in AB is different than in our formulation as firms first make announcements and then after observing these announcements they decide on prices and maximum market shares. Thus strategies are defined as $m(J_{i,t})$ and $x(J_{i,t}, m_{i,t}, m_{-i,t})$.

In order to fit the AB model into our setup there are three modifications that need to made. The first two are simple to make. The third one is a more fundamental modification of the structure of the model.

(i) The quantities and the market shares should be taken from a finite set. We do not think that such a modification dramatically change the model.

(ii) For convinience we assume in our formulation that all the decision are taken simultaneously. In the AB model each period the firms first make an announcement regarding their cost position and after observing the announcements of all the firms they make their price/market share decisions. We can modify our formulation to include sequential decision making within a period. But such a modification will be at the cost of having a larger state space.[20]

[18] We changed the notation in AB for ease of presentation.

[19] In the general vase the firm's private information is the history of their cost position, like in our formulation AB also do not condition startegies on such an history.

[20] In order to modify our setting we need to "break" every period into two. First, for the announcement decision we need to define value functions for every combination of

(iii) In the AB model the set of public information is defined by all the history of the publicly observed variables and as such it is not finite. This is probably the major obstacle in implementing our formulation. A possible way to ensure finite state space is to assume imperfect recall such that players cannot remember more than k period. Alternatively, one can assume that cost positions are revealed every k periods, for example due to a either a physical or a regulatory constraint.

Signalling and Maintenance.

A more realistic model would allow a firm to chose from a finite menu of bid functions for each of its generators. If the bid functions are public then the bids become informationally relevant signals of the cost state of the generators.[21] That is if we let

$$M_i = \Pi_j\{0, m_{i,1,j}(\cdot), \ldots m_{i,K,j}(\cdot)\}$$

be the set of possible bids for firm i, then $m_{i,t} = [m_{i,1,t}, \ldots m_{i,n_i,t}] \in M_i^{n_i}$ can provide a signal on the cost states of the generators of firm i, and hence are informationally relevant for period $t+1$.

Indeed the point to note here is that without further restrictions each $m_{i,t}$ is needed for the information set in *all* future t. This despite the fact that by period $t + \overline{\tau}$ each of the generators will have been maintained and their ω reset to zero. Consequently without further restrictions our model and computational techniques do not apply.

It is easiest to see this in a simple example in which there are two firms, each with a single generator, and two possible bids besides the maintenance decision, so $(0, m_1, m_2) = M_i$. Now assume that firm 1 did maintenance on its generator in period 3 while firm 2 did maintenance on its generator in period 5. Consider the informationally relevant variables for firm 2 in period 6. Firm 2's bid will depend on its belief on firm 1's cost. As a result firm 1's bid in both period four and five are informationally relevant in period 6. However the bids of firm 1 in these periods were a function of the informationally

public information, private cost position and all possible announcement of the single firm. Then for the pricing and market share decision there are value function for every public information, private cost position, vector of observed announcements of the firms and all possible combinations of prices and market shares for the single firm.

[21] If the bid functions were private information then price becomes an informationally relevant signal and a similar set of issues to those discussed below would arise.

relevant variables at that time, which, among other variables, included firm 2's bid in period 2. So firm 2's bid in period 2 must be used to interpret firm 1's bid in period 3, and period 2 is prior to the resetting of both firm's cost. This reasoning can be iterated to show that the whole history of bids is, in general, informationally relevant for all periods.

There are a number of ways in which these histories can be truncated and different reasons for truncation are likely to be relevant for different applied problems. One possiblity is that there is a mechanism which periodically reveals all payoff relevant random variables simultaneoulsy, and makes them common knowledge. This will imply that all the bids prior to the period in which all payoff relevant variables are made common knowledge are informationally irrelevant thereafter. In the current example this would occur if there was a periodic simutaneous inspection of all generators. A similar mechanism was adopted by Fershtman and Pakes (2005) in a model of collusion in a dynamic game with asymetric information (in which case cartels periodically met to share information).

Alternatively we could truncate a history if there was a sequence of bids that fully revealed all the payoff relevant random variables (in our case the costs of all the generators). More formally the condition of full revelation in a time period is that conditional on the public information available at that time, each players' optimal action fully reveal its type. For the game to be a finite state game we require, in addition, that fully revealing states are visited infinitely often.

A different approach would be to assume bounded memory, that is assume that only the information which accumulates over the immediately preceeding K periods can be stored by the agent and used to form strategies, as in the literature

Note that this restriction does not imply that the history prior to period $t-k$ is informationally irrelevant in period t. Rather it assumes that players do not condition strategies on that information. This would again put us back into a model with a finite state space.

5 Practical Issues in Computing the Equilibrium.

Even in our simple model we run into RAM constraints for memory, and swapping is very time intensive. Procedures for overcoming these problems.

- Storage first by public information and then within the common public, by private information (using say separate hashing function).
- Start with a small initial state space but high initial values on that state space to allow for experimentation.
- Keep in memory a number for the last iteration at which each J_i was visited.
- After "y" million iterations, drop from memory all J_i which were not visited in the last "x" million iterations.
- Use the equilibrium values from the small model as initial conditions for a larger model. In particular say we start with two generators for firm, and maximum demand. Output looks like $W(\eta, m|\omega_1, \omega_2, \tau_{11}, \tau_{12}, \tau_{21}, \tau_{22})$ now use these $W's$ to initialize $W^0(\eta, m|\omega_1, \omega_2, \omega_3, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{21}, \tau_{22}, \tau_{2,3})$ by chosing the W from the smaller dimensional model for the J_i with
  - Among $\omega_1, \omega_2, \omega_3$, chose the lowers two ω's
  - Among $\tau_{11}, \tau_{12}, \tau_{13}$ chose the lowest two τ
  - Among $\tau_{21}, \tau_{22}, \tau_{23}$ chose the highest two τ

6 Computational Comparison.

Standard iterative computational algorithms for equilibrium fixed points keep values and policies in memory for each point in the state space, update these values sequentially at every iteration, and stop the algorithm when the values and policies in two successive iterations are equal. Their computational burden is determined as the product of; (i) the number of points in the state space, (ii) the burden of updating each point at each iteration, and (iii)the number of iterations (for more detail, and an extensive discussion of procedures for simplifying the computational burden, see Judd, 1998). As noted in

Pakes and McGuire (2001), the use of stochastic approximation algorithms reduces the burden of computing Markov Perfect equilibrium in dynamic games in two ways: the algorithm eventually focuses in on a recurrent class of points thus reducing the number of points updated, and the computational burden at each point evaluated changes from the burden involved in computing integrals over future values to the burden involved in updating averages of past values.

When we allow for asymmetric information the computational burden updating each point in an iterative algorithm increases *dramatically*. The expectation over the future required for the updated continuation value is taken with a probability distribution which itself must be computed as a ξ-fold convolution of more primitive distributions, and each of these distributions depends on the policies relevant at *alternative* points in the state space. The fact that those probabilities depend on policies at other states will require us to either increase the memory for each state substantially, or to search and retrieve information from different states each time we update for a particular state. Given these policies the computational complexity of the ξ-fold convolution increases exponentially in ξ. In rather stark contrast, the updating burden of the stochastic algorithm remains the same; it still only need to update averages.

6.1 The Test Statistics.

Recall from section 2.2 that the test uses the policies outputted by the algorithm to simulate sample paths, and then compares the average value of the simulated paths at each point to the values outputted by the algorithm (after accounting for sampling error in the simulated paths). The test statistic itself has the interpretation of an $\mathcal{L}^2(P)$ norm in the percentage deviation between the simulated and estimated values, where the weights in P are determined by the frequency with which the different points are visited (as an approximation to their probability in the invariant distribution on the recurrent class of points).

As noted by varying the number of periods simulated or the stopping rule (the τ in section 2.2) we can construct many different tests, and the test with $\tau \equiv 1$ is the stochastic analogue of the stopping rule typically used in iterative algorithms. Figure 1 graphs the values of both the $\tau = 1$ test, and a second test which does not rely on simulation but is not generally available for dynamic games with assymetic information. In our example

there are a subset of the points in $\mathcal{R}$, the points at which there is a meeting (or $\xi = 0$), in which there is no asymmetric information. At these points we can calculate the discounted future values that the algorithm's policies imply without computing posterior distributions, and hence we can mimic the exact test (i.e. a test that does not require simulation) used in models without assymetric information (see Pakes and McGuire,2001, and the literature they cite). The solid line provides the $\mathcal{L}^2(P)$ norm of the percentage differences of the exact test at the points with full information[22].

Each test statistic was computed for ten million iteration intervals, starting at the one hundred millionth iteration, and going to iteration one billion one hundred million. The solid line in the figure shows that the exact test statistic declines rather rapidly between the ten and one-hundred and thirty millionth iterations, falls further, but at a slower pace, until about the 350 millionth iteration, and essentially flatten out at about .0006 (or .06%) thereafter (they do decline further, but at almost an imperceptible rate).

This test does not provide information on the closeness of the estimated values to the values implied by our policies at points at which there is no meeting. The second test, which is exactly the test described in section 2.2 (with $\tau = 1$), uses simulation to construct comparisons for all points with weights given by the points' frequencies in the last ten million draws. These results are more "jumpy" (as one would expect from a stochastic test), tend to fall continually until about eight hundred million iterations, but then flatten out at between .002 and .0026 (or .2 to .26%)[23].

We view these results as indicating we have a close enough approximation for our purposes, and now move on to consider their implications.

[22] In fact we computed several other tests, but we will suffice here with a description of how these two test statistics behaved as the results on the other tests were similar.

[23] The fact that the norm is larger for the second test could result from a number of different phenomena. There could be; a poorer fit at the points where there is assymetric information, some inaccuracy generated by the random number generator, or noise in our measure of sampling variances. Two other points are worth noting here. First these are uncenterred differences. The actual correlation between the two discounted values was noticeably higher, with values exceeding .99999. Second we did not worry about boundary points in the tests for two reasons. First they were hit very infrequently, and second we initiate all points at very high values. As a result the points connected to the points on the boundary tend to be assigned higher values than their equilibrium values.

7 Concluding Remark

We have presented a simple algorithm for computing an intuitive notion of MPE for finite state dynamic games with asymmetric information. The algorithm is relatively efficient in that it does not require; storage and updating of posterior distributions, explicit integration over possible future states to determine continuation values, or storage and updating of information at all possible points in the state space. To illustrate our algorithm we computed the equilibrium of oligopolistic industries with collusive interactions. This showed that parameters determining information flows can effect market structure and through market structure producer and consumer surplus. More generally our hope is that the framework for analyzing dynamic games with assymetric information presented here will enable more realistic analysis of interactions among economic agents.

References

Athey S and Bagwell, K (2004) "Collusion with Persistent Cost Shocks" *Econometrica*, (forthcoming).

Bajari, Benkard and Levin ???

Beertsekas, D. and J. Tsikilis (1996) *Neoro-Dynamics Programming.* Belmont, MA: Athena Scientific Publications.

Besanko, D. and U. Doraszelski (2002) "Capacity Dynamics and Endogenous Asymmetries in Firm Size" *mimeo*, Northwestern University.

Cheong K.S. and K. Judd, (2003) "Mergers and Dynamic Oligopoly," *Journal of Economic Dynamics and Control*, forthcoming.

Dekel, E., D. Fudenberg and D.K. Levine (2004), "Learning to play Bayesian Games" *Games and Economic Behavior*, 46, 282-303.

Doraszelski, U. and M. Satterthwaite, (2003) "Foundations of Markov-Perfect Industry Dynamics: Existence, Purification, and Multiplicity," mimeo Hoover Institution.

Ericson R. and A. Pakes, (1995) "Markov-Perfect Industry Dynamics: A Framework for Empirical Work" *Review of Economic studies*, 62, 53-82.

Fershtman C. and A. Pakes (2000), "A Dynamic Oligopoly with Collusion and Price Wars" *The Rand Journal of Economics*, 31(2) pp. 207-236.

Freedman, D. (1983) *Markov Chains.* New York: Springer Verlag.

Fudenberg, D and D. Kreps (1994) "Learning in Extensive Form Games. II Experimentation and Nash Equilibrium" Mimeo Stanford University.

Fudenberg, D. and D.K. Levine (1993a) "Self Confirming Equilibrium" *Econometrica*, 61(3), 523-545.

Fudenberg, D. and D.K. Levine (1993b) "Steady State Learning and Nash Equilibrium" *Econometrica*, 61(3), 547-573.

Gowrisankaran, G. (1999) "A dynamic model of endogenous horizon mergers" *The Rand Journal of Economics* 71, pp. 51-63.

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

Judd, K., J. Conklin and S. Yeltekin (2003) "Computing Supergame Equilibria," *Econometrica* 71, 1239-1254.

Markovich S. (2000), "Snowball - The Evolution of Dynamic Oligopolies with Network Externalities," mimeo, Tel-Aviv University.

Maskin, E. and J. Tirole (1988), "A Theory of Dynamic Oligopoly, I: Overview and Quantity Competition with Large Fixed Costs" *Econometrica*, 56, 549-570.

Maskin, E. and J. Tirole (2001), "Markov Perfect Equilibrium: Observable Actions", *Journal of Economic Theory*, 100, pp. 191-219.

Pakes, A. (1994); "Dynamic Structural Models: Problems and Prospects", Chapter 5 in C. Sims (ed.) *Advances in Econometrics Sixth World Congress*, Vol. II, pp 171-260, Cambridge University Press.

Pakes, A. and P. McGuire (1994), "Computing Markov Perfect Nash Equilibrium: Numerical Implication of a Dynamic Differentiated Product Model", *The Rand Journal of Economics*, 25, pp. 555-589.

Pakes, A. and P. McGuire (2001), "Stochastic Algorithms, Symmetric Markov Perfect Equilibrium, and the "Curse" of Dimensionality", *Econometrica* pp.1261-81.

Pakes Ostrovsky and Berry...

Robbins and Monroe (1954); "A Stochastic Approximation Technique", *Annals of Mathematics and Statistics*,.